

# Desenvolvimento Web

## Sumário

Introdução.....	2
Introdução aos Servidores.....	3
1 - Introdução aos Servidores web e SGBD.....	3
1.1 - Remover um serviço do Windows.....	3
1.2 - Quebrar senha do Windows com o Hiren Boot.....	3
1.3 - Captura de Tela no Mint.....	3
1.4 - Gravador de Imagem em Pendrive no Linux Mint.....	3
1.5 - Desativando Hibernação no Windows 8+.....	4
1.6 - Permissões no Diretório Web Linux.....	4
1.7 - Conectar ao Nemo ou Nautilus.....	5
1.8 - Instalar pacotes do Servidor Web.....	5
1.9 - Virtual Host no ubuntu.....	6
1.10 - VirtualHost no Xampp.....	8
2 – Introdução ao PostgreSQL.....	10
2.1 - Usuários e Privilégios.....	13
3 - Introdução ao HTML.....	14
4 - Introdução do JavaScript.....	21
5 - Introdução ao CSS.....	23
6 - Introdução ao XML.....	26
7 - Introdução ao jQuery.....	28
8 - Introdução do Bootstrap.....	29
9 - Ambiente de Desenvolvimento.....	30
10 - Introdução ao PHP.....	31
11 - Introdução ao PHP Orientado a Objetos.....	40
12 - Introdução ao PDO.....	42
13 - Introdução ao MySQL.....	44
14 - CMS Joomla.....	49
14.1 - Instalação do CMS Joomla 3.6.5.....	49
14.2 - Gerenciando Template para Joomla com Gantry.....	52
14.3 - Gerenciamento de Conteúdo do Joomla 3.6.5.....	61
14.4 - Administração do Joomla 3.6.5.....	68
15 - Framework CakePHP 3.....	73
15.1 - Convenções no CakePHP 3.....	73
15.2 - Instalação do CakePHP 3.....	77
15.3 - Usando o Plugin cake-control-br do CakePHP 3.....	82
15.4 - Arquivos em Geral do CakePHP 3.....	85
15.5 - Gerando Código com Bake no CakePHP 3.....	86
15.6 - Dicas.....	92
15.7 - Projeto com GitHub e Packagist.....	96
15.8 - Implementação de busca no CakePHP 3.....	104
15.9 – MVC no CakePHP 3.....	106
15.10 - Form Helper.....	119
15.11 - Um Resumo da codificação do CakePHP.....	128
15.12 - Comunicação entre Model, Controller e View.....	137
15.13 - Customizar paginação.....	138
15.14 - Acesso a Banco de Dados.....	139
15.15 - Usando displayField().....	140

# Introdução

Um e-book meio atípico, pois aborda de servidor à Joomla 3, passando pelos conhecimentos html, mysql, php, cakephp 3 entre outros, mas tudo isso somente por ser apenas resumidas receitas de bolo.

Autor: Ribamar FS – ribafs

<https://ribafs.github.io/#sobre>

<http://ribafs.org/portal/sobre-mim>

Fortaleza, 25 de abril de 2017

## **Guia de Consulta Prático**

Este livro tem a pretensão de ser um guia de consulta para desenvolvedores web, com receitas práticas e prontas sobre boa parte dos assuntos de interesse.

# Introdução aos Servidores

## 1 - Introdução aos Servidores web e SGBD

### 1.1 - Remover um serviço do Windows

Execute:

```
sc delete nomedoserviço
```

```
sc delete apache2
```

```
sc delete mysql
```

### 1.2 - Quebrar senha do Windows com o Hiren Boot

Dê o boot e escolha Mini Windows XP Ao iniciar o CD abra o programa "HBCD Menu" localizado na área de trabalho: Programs > Passwords / Keys > Windows Login > NTPWEdit (Reset Xp/Vista/7 User Passowrd)

<http://www.hirensbootcd.org>

### 1.3 - Captura de Tela no Mint

Com imagem e som

```
sudo apt-get install vokoscreen
```

### 1.4 - Gravador de Imagem em Pendrive no Linux Mint

- Iniciar

- Gravador de Imagem USB

## 1.5 - Desativando Hibernação no Windows 8+

Quem tem windows e Linux na mesma máquina, com Windows 8, 10 ou superior

Pressione a Tecla **Windows + X** e execute como Administrador o **Prompt de comando**.

Com o Prompt aberto digite:

```
powercfg -h off
```

e pressione enter.

Agora a hibernação no Windows 8 esta desativada.

## 1.6 - Permissões no Diretório Web Linux

Criar um script

```
sudo nano /usr/local/bin/perms
```

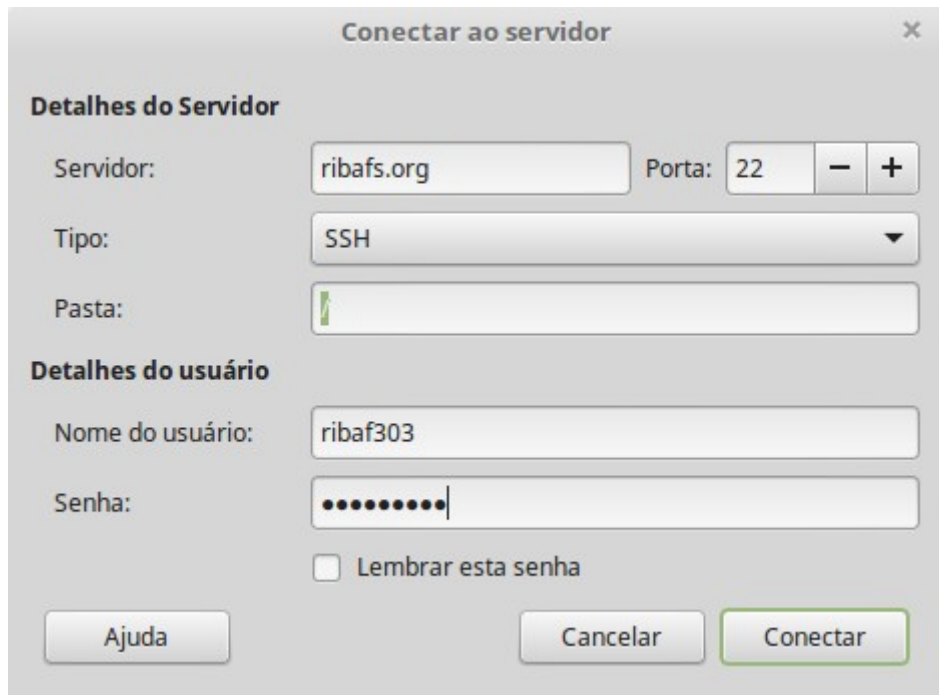
```
#!/bin/sh
clear;
# Permissões para uso em desktop
echo "Aguarde enquanto configuro as permissões do /var/www/html/$1";
echo "";
# addgroup webdevel
chown -R www-data:webdevel /var/www/html/$1;
chmod -R g+s /var/www/html/$1;
find /var/www/html/$1 -type d -exec chmod 2775 {} \;
find /var/www/html/$1 -type f -exec chmod 2664 {} \;
echo "";
echo "Concluído!";
```

```
sudo chmod 700 /usr/local/bin/perms
```

Executar

```
sudo perms
```

## 1.7 - Conectar ao Nemo ou Nautilus



## 1.8 - Instalar pacotes do Servidor Web

```
sudo su
```

```
apt-get -y install aptitude unzip ntp ntpdate git;
```

```
apt-get -y install apache2 libapache2-mod-php7.0;
```

```
a2enmod rewrite;
```

```
apt-get -y install postgresql postgresql-contrib;
```

```
apt-get -y install mysql-server;
```

```
apt-get -y install php7.0 mcrypt php7.0-mcrypt php7.0-gd php7.0-mysql php7.0-sqlite3 php7.0-pgsql php7.0-ldap;
```

```
apt-get -y install php7.0-mcrypt php-pear php7.0-xsl curl php7.0-curl phpunit php-xdebug php7.0-intl composer;
```

```
apt-get -y install php7.0-zip php7.0-mbstring php-gettext php-mbstring php7.0-fpm php-auth phpmyadmin;
```

```
apt-get -y install php-apcu;
```

```
wget http://ftp.ussg.iu.edu/linux/ubuntu/pool/main/m/memcached/memcached_1.4.25-2ubuntu1_amd64.deb;
```

```
dpkg -i memcached_1.4.25-2ubuntu1_amd64.deb;
```

```
apt-get -y install php-memcache;
```

Suporte a mod\_rewrite e .htaccess no Apache 2.4 trocando None por All

nano /etc/apache2/apache2.conf:

```
<Directory />
    Options Indexes FollowSymLinks Includes ExecCGI
    AllowOverride All
    Order deny,allow
    Allow from all
</Directory>
```

## 1.9 - Virtual Host no ubuntu

<https://www.vivaolinux.com.br/topico/Apache-Web-Server/virtualHost-1>

Muito útil para quem precisa executar um site no raiz web e que precisa de vários sites, todos no raiz.

**Sudo nano /etc/hosts**

```
127.0.0.1 ribafs.loc www.ribafs.loc
```

```
127.0.0.1 local.loc www.local.loc
```

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/ribafs.conf
```

**Apagar tudo o existente e adicionar:**

```
sudo nano /etc/apache2/sites-available/ribafs.conf
```

```
<VirtualHost *:80>
ServerAdmin ribafs@gmail.com
ServerName www.ribafs.loc
ServerAlias ribafs.loc
DirectoryIndex index.php
DocumentRoot /var/www/ribafs
LogLevel warn
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
<Directory /var/www/ribafs/>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
    DirectoryIndex index.html index.php
</Directory>
</VirtualHost>
```

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/local.conf
```

## Apagar tudo e adicionar:

```
sudo nano /etc/apache2/sites-available/local.conf
```

```
<VirtualHost *:80>
ServerAdmin ribafs@gmail.com
ServerName www.local.loc
ServerAlias local.loc
DirectoryIndex index.php
DocumentRoot /var/www/local
LogLevel warn
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
<Directory /var/www/local/>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
    DirectoryIndex index.html index.php
</Directory>
</VirtualHost>
```

```
sudo a2ensite ribafs
sudo a2ensite local
service apache2 reload
service apache2 restart
```

Lembrando que já temos o localhost, que é em torno de:

```
<VirtualHost *:80>
ServerAdmin ribafs@gmail.com
ServerName www.localhost
ServerAlias localhost
DirectoryIndex index.php
DocumentRoot /var/www/html
LogLevel warn
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
<Directory /var/www/html/>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
    DirectoryIndex index.html index.php
</Directory>
</VirtualHost>
```

## Para o caso de Desinstalar o local

```
sudo a2dissite local  
service apache2 restart
```

## 1.10 - VirtualHost no Xampp

<http://www.pauloacosta.com/2016/07/criando-multiplos-virtual-hosts-no-xampp/>

Editar com o Notepad++ com Administrador o arquivo:

```
c:\windows\system32\drivers\etc\hosts
```

Adicione:

```
127.0.0.1    local.loc  www.local.loc  
127.0.0.1    ribafs.loc www.ribafs.loc
```

### Criar as pastas

```
c:\xampp\htdocs\ribafs
```

```
c:\xampp\htdocs\local
```

### Editar

```
c:\xampp\apache\conf\extra\httpd-vhosts.conf
```

E adicionar ao final:

```
<VirtualHost *:80>  
    ServerName default  
    DocumentRoot "C:\xampp\htdocs"  
    ErrorLog "logs/default-error.log"  
    CustomLog "logs/default-access.log" common  
    <Directory "C:\xampp\htdocs">  
        DirectoryIndex index.php index.html index.htm  
        AllowOverride All  
        Order allow,deny  
        Allow from all  
    </Directory>  
</VirtualHost>  
  
<VirtualHost *:80>  
    ServerName local.loc  
    ServerAlias www.local.loc  
    DocumentRoot "C:\xampp\htdocs\local"
```



```
ErrorLog "logs/dominio1-error.log"
CustomLog "logs/dominio1-access.log" common
<Directory "C:\xampp\htdocs\local.loc">
    DirectoryIndex index.php index.html index.htm
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>
</VirtualHost>
```

```
<VirtualHost *:80>
    ServerName ribafs.loc
    ServerAlias www.ribafs.loc
    DocumentRoot "C:\xampp\htdocs\ribafs"
    ErrorLog "logs/ribafs-error.log"
    CustomLog "logs/ribafs-access.log" common
    <Directory "C:\xampp\htdocs\ribafs">
        DirectoryIndex index.php index.html index.htm
        AllowOverride All
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

## **Reiniciar o Apache**

## 2 – Introdução ao PostgreSQL

```
apt-get update
```

```
apt-get install postgresql postgresql-doc postgresql-9.5-slony1 postgresql-9.5-postgis
```

### **Permitir Acesso somente Local**

Para permitir que aplicativos e sites criados com o PHP tenham acesso através de uma conexão onde usam 'host=localhost', ou seja, local, devemos executar como root.

### **Criar bancos**

Criar usuários com direitos somente ao banco

Cada site deve ter um usuário exclusivo e um banco onde tem plenos poderes.

### **CRIAR MAIS DOIS USUÁRIOS**

Criar usuários que não podem criar usuários, bancos nem roles

```
su - postgres
```

```
createuser ribafs
```

```
createuser joao
```

### **Alterar a senha dentro do SGBD**

```
psql
```

```
ALTER USER postgres WITH PASSWORD 'senha';
```

```
\q      para sair
```

```
cp /etc/postgresql/9.1/main/postgresql.conf /etc/postgresql/9.1/main/postgresql.confCOP
```

```
nano /etc/postgresql/9.1/main/postgresql.conf
```

```
listen_addresses = '*' (Escutar todas os IPs. Controlar no pg_hba.conf)
```

Neste arquivo (pg\_hba.conf) você consegue restringir o acesso ao seu banco de dados por IP, por usuário, por tipo de senha, etc.

```
cp /etc/postgresql/9.1/main/pg_hba.conf /etc/postgresql/9.1/main/pg_hba.confCOP
```

```
nano /etc/postgresql/9.1/main/pg_hba.conf
```

## Liberar o IP do sysadmin (10.40.0.103)

```
# Database administrative login by UNIX sockets
local all postgres ident (Usuário postgres usa a mesma senha do SO)
```

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
```

```
# "local" is for Unix domain socket connections only
```

```
local all all ident
```

```
# IPv4 local connections:
```

```
host all all 127.0.0.1/32 md5
```

```
host all all 10.40.0.133/32 md5
```

Toda uma rede:

```
host all all 10.0.0.0/24 md5
```

```
exit
```

```
/etc/init.d/postgresql restart
```

Para não atribuir senha para o postgres, usando somente através do root podemos usar:

```
su postgres -c psql postgres
```

Alguns scripts em

```
/usr/share/postgresql/8.4/
```

## Uso Remoto do PostgreSQL

Acesso remoto do IP 192.168.1.67

No postgresql.conf listen\_addresses deve estar com '\*'

Alterar o pg\_hba.conf na linha do IP para trust

```
host all all 192.168.1.67/32 md5
```

Para toda a rede:

```
host all all 192.168.1.0/24 md5
```

Restartar

```
/etc/init.d/postgresql restart
```

Acessar com:

```
psql -h IP -U usuario -d banco
```

ou

```
psql -h IP -U usuario
```

Listar bancos remotamente

```
psql -l -h 192.168.1.12 -U postgres
```

Ajuda

```
psql --help
```

## Backup local e restore remoto

```
pg_dump banco | psql -h hostname banco -U postgres
```

```
pg_restore portal -f
```

## Máscara Explicada

Apenas para iluminar um pouco, aquele número que vem depois da barra "/" significa o número de bits que ele vai utilizar na máscara. Vejamos.

Suponha que você deixe 189.0.0.0/24, o que vai acontecer?

1) Você vai tentar conectar no IP do seu servidor a partir de seu IP de origem IP 2) Seu servidor vai pegar o seu IP de origem e fazer um cálculo de máscara usando 24 bits, numa comparação XOR bit-a-bit que vai resultar em 189.22.33.0

3) Ele vai pegar o resultado do cálculo acima e comparar com seu arquivo e vai identificar que 189.22.33.0 NÃO É IGUAL A 189.0.0.0.

Portanto você tem que utilizar 189.0.0.0/8, pois assim ele vai pegar o seu IP IP, vai fazer uma comparação XOR bit-a-bit e vai ter como resultado 189.0.0.0, com esse resultado ele vai comparar com o 189.0.0.0 e vai reconhecer a IGUALDADE entre eles e vai aceitar.

## Em resumo.

```
IP/8 => IP/255.0.0.0 = 192.0.0.0
```

```
IP/16 => IP/255.255.0.0 = 192.168.0.0
```

```
IP/24 => IP/255.255.255.0 = 192.168.1.0
```

```
IP/32 => IP/255.255.255.255 = 192.168.1.12
```

Outras máscaras são possíveis através de deslocamento de bit do parte da rede para o host, obtendo-se sub-redes, mas aí já acabamos fungindo do escopo da lista.

--

Dickson S. Guedes

## Criar Usuários com privilégios Restritos

Permitir Acesso somente Local

Para permitir que aplicativos e sites criados com o PHP tenham acesso através de uma conexão onde usam 'host=localhost', ou seja, local, devemos executar como root.

Neste arquivo (pg\_hba.conf) você consegue restringir o acesso ao seu banco de dados por IP, por usuário, por tipo de senha, etc.

```
cp /etc/postgresql/8.4/main/pg_hba.conf /etc/postgresql/8.4/main/pg_hba.confCOP
```

```
nano /etc/postgresql/8.4/main/pg_hba.conf
```

Liberar o IP do servidor WEB (10.10.0.102) e do servidor de Email (10.10.0.103)

```
# Database administrative login by UNIX sockets
```

```
local all postgres ident (Usuário postgres usa a mesma senha do SO)
```

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all ident
# IPv4 local connections:
host all all 127.0.0.1/32 md5
host all all 10.10.0.102/32 md5
#host all all 10.10.0.103/32 md5

Toda uma rede:
host all all 10.0.0.0/24 md5
```

```
/etc/init.d/postgresql restart
```

## 2.1 - Usuários e Privilégios

```
sudo su
```

```
su - postgres
```

```
createdb joomlapg
```

```
createuser joomlapg
```

```
psql joomlapg
```

```
ALTER DATABASE joomlapg OWNER TO joomlapg;
```

```
alter role joomlapg password 'joomlapg';
```

```
REVOKE CONNECT ON DATABASE joomlapg FROM PUBLIC;
```

```
GRANT CONNECT ON DATABASE joomlapg TO joomlapg;
```

```
---
```

```
REVOKE ALL ON ALL TABLES IN SCHEMA public FROM PUBLIC;
```

```
GRANT ALL ON ALL TABLES IN SCHEMA public TO joomlapg;
```

## 3 - Introdução ao HTML

**HTML** (abreviação para a [expressão inglesa](#) *HyperText Markup Language*, que significa *Linguagem de Marcação de Hipertexto*) é uma [linguagem de marcação](#) utilizada na construção de [páginas na Web](#). Documentos HTML podem ser interpretados por [navegadores](#). A [tecnologia](#) é fruto da junção entre os padrões [HyTime](#) e [SGML](#).

Toda página web é criada com HTML. Pode ter algo mais mas não pode faltar HTML.

HyTime é um padrão para a representação estruturada de hipermídia e conteúdo baseado em tempo. Um documento é visto como um conjunto de eventos concorrentes dependentes de tempo (como áudio, vídeo, etc.), conectados por hiperligações. O padrão é independente de outros padrões de processamento de texto em geral.

SGML é um padrão de formatação de textos. Não foi desenvolvido para hipertexto, mas tornou-se conveniente para transformar documentos em hiper-objetos e para descrever as ligações.

(da Wikipédia)

### Escola Online

<http://www.w3c.br/Cursos/CursoHTML5>

<https://www.w3schools.com/html/>

<https://www.caelum.com.br/apostila-html-css-javascript/>

### O HTML5 é a nova versão do HTML.

HTML significa Hyper Text Mark-up Language, linguagem de marcação de texto. Uma linguagem criada para mostrar texto em páginas mas que evoluiu muito e atualmente faz praticamente tudo que outras linguagens mais poderosas fazem. É conhecimento obrigatório para qualquer desenvolvedor web.

O HTML é uma linguagem de marcação, que usa tags/etiquetas para o reconhecimento do que fazer. Todos os sites na Internet usam o HTML para que sejam exibidos por um navegador.

A maioria das tags tem a tag inicial e a de fechamento, veja exemplo abaixo.

### O foco do HTML é o conteúdo.

Detalhes:

[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

Uma página básica e simples em HTML5 deve conter as tags das 2 primeiras linhas:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta charset="UTF-8">
    <title>Título do site</title>
  </head>
```

```
<body>
  <h1>Esta é a parte principal da página</h1>
  <p>Aqui um texto normal</p>
</body>
</html>
```

**Para ser automaticamente reconhecido** pelos navegadores deve ter a extensão .htm ou .html.

Acima para identificar HTML5 adiciona-se mais uma tag no início do arquivo:

Veja que adicionamos a primeira e mudamos a segunda (<html>).

Para a criação de um texto em HTML precisamos de um editor de texto e de um navegador. Tanto podemos usar um editor simples como o bloco de notas, o notepad++ já é melhor e melhor ainda é ma IDE como o NetBeans.

Se criarmos um arquivo chamado teste.html contendo:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
  </head>
  <body>
  </body>
</html>
```

Ao abrir num navegador não veremos nada, apenas uma página em branco.

Apenas algumas tags mostram conteúdo, outras são apenas estruturais, como html, head, body, etc.

## Validação

O site do w3C tem uma ferramenta da validação online.

Acesse

<http://validator.w3.org/>

Clique em Validate by Direct input

Cole o código que deseja testar e clique em Check.

Ele fará uma avaliação do seu código, mostrando os erros.

Veja que ele apresenta mais duas alternativas de validação: URL e arquivo.

O HTML5 trouxe muitos recursos importantes para os sites, tanto para si quanto para seus parceiros CSS e JavaScript. Basta ver a quantidade de ferramentas que existem e que aparecem a cada dia para a criação de games para mobile.

## Comentários

Em HTML para comentar um trecho de código usamos:

```
<!-- Aqui todo o código
será ignorado pelo navegador.
Ou seja, temos um comentário para
múltiplas linhas.
-->
```

## Dica sobre o Notepad++

Para trabalhar com caracteres UTF-8 e salvar assim:  
Formatar - Codificação em UTF-8 (sem BOM)

## Exemplos

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta charset="UTF-8">
    <title>Título do site</title>
  </head>
  <body>
    <h1>Título nível 1</h1>
    <h2>Título nível 2</h2>

    <div>Aqui temos uma divisão lógica</div>

    <hr>
    Acima mostrarei uma linha horizontal.

    <ol>
      <li>Aqui é um item de uma lista ordenada (ol - ordered list)</li>
      <li>li significa list item</li>
      <li>Podemos adicionar tantos itens quantos quisermos</li>
    </ol>

    <ul>
      <li>Aqui é um item de uma lista não ordenada (ul)</li>
```



```
<li>li significa list item</li>
<li>Podemos adicionar tantos itens quantos quizermos</li>
</ol>

<p>Aqui uma <strong>palavra em negrito</strong> em <br>
quebra de linha</p>
</body>
</html>
```

## Exibindo Imagens

```

```

## Links

```
<a href="http://ribafs.org" target="_blank">RibaFS Portal</a>
target="_blank" - para abrir a página noutra aba ou janela.
```

## Podemos linkar para IDs da página, veja:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Título do site</title>
  </head>
  <body>
    <div id="topo">Topo da página</div>
    <h1>Título nível 1</h1>
    <h2>Título nível 2</h2>

    <div>Aqui temos uma divisão lógica</div>

    <hr>
    Acima mostrarei uma linha horizontal.

    <ol>
      <li>Aqui é um item de uma lista ordenada (ol - ordered list)</li>
      <li>li significa list item</li>
      <li>Podemos adicionar tantos itens quantos quizermos</li>
    </ol>

    <ul>
```

```
<li>Aqui é um item de uma lista não ordenada (ul)</li>
<li>li significa list item</li>
<li>Podemos adicionar tantos itens quantos quisermos</li>
</ol>
```

### Link para ID

```
<a href="#topo">Voltar ao topo</a>
</body>
</html>
```

Crie um arquivo chamado links.html e cole o conteúdo acima para testar o link para a mesma página.

Precisará adicionar mais texto no body para que possa rolar a tela e clicar no link que rolará a tela para o topo.

### Link para e-mail

```
<a href="mailto:ribafs@gmail.com?Subject='E-mail de teste'">Contato</a>
```

### Tabelas

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Título do site</title>
  </head>
  <body>
    <div id="topo">Topo da página</div>
    <h1>Título nível 1</h1>

    <table>
      <caption>Minha Tabela HTML5</caption>
      <tr><td>Celula 11</td><td>Célula 12</td></tr>
      <tr><td>Celula 21</td><td>Célula 22</td></tr>
      <tr><td>Celula 31</td><td>Célula 32</td></tr>
    </table>
  </body>
</html>
```

## Formulários

Neste assunto o HTML5 trouxe muitas novidades interessantes.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>--Formulário Contato--</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <link href="estilo.css" rel="stylesheet" media="all" />
  <script src="script.js"></script>
</head>
<body>
  <form name="meu_form">
    <h1>Entre em contato</h1>
    <p class="nome">
      <input type="text" id="nomeid" placeholder="Tiago Vale" required="required"
name="nome" />
      <label for="nome">Nome</label>
    </p>
    <p class="fone">
      <input type="text" id="foneid" placeholder="(xx)xx-xx-xx-xx" name="fone" />
      <label for="fone">Fone</label>
    </p>
    <p>
      <input type="email" id="emailid" placeholder="fulano@mail.com" name="email" />
      <label for="email">Email</label>
    </p>
    <p>
      <textarea placeholder="Deixe sua opinião"></textarea>
    </p>
    <p class="submit">
      <input type="submit" onclick="Enviar();" value="Enviar" />
    </p>
  </form>
</body>
</html>
```

Veja em CSS e JavaScript os arquivos estilo.css e script.js.

## Algumas novas TAGs do HTML5

```
<header>Texto</header>
```

```
<footer>Texto</footer>
```

```
<nav>Texto</nav>
```

```
<section>Texto</section>
```

```
<section>
```

```
  <article>
```

```
    Texto
```

```
  </article>
```

```
  <article>
```

```
    Texto
```

```
  </article>
```

```
</section>
```

E muitos outros. Veja aqui:

[https://developer.mozilla.org/pt-BR/docs/Web/HTML/HTML5/HTML5\\_element\\_list](https://developer.mozilla.org/pt-BR/docs/Web/HTML/HTML5/HTML5_element_list)

## **ídeos**

```
<video width="640" height="480" controls>
```

```
  <source src="video.mp4" type="video/mp4">
```

```
  <source src="video.webm" type="video/webm">
```

```
  Meus Vídeos
```

```
</video>
```

## **Áudio**

```
<audio controls>
```

```
  <source src="audio.mp3" type="audio/mpeg">
```

```
  <source src="video.ogg" type="audio/ogg">
```

```
  Meus Áudio
```

```
</audio>
```

Algumas tags foram adicionadas pelo HTML5 e outras foram removidas. Veja aqui:

<http://newhtml5.blogspot.com.br/2011/03/novas-tag.html>

<http://www.scriptcase.com.br/blog/objetos-removidos-html5/>

[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

## 4 - Introdução do JavaScript

*JavaScript* é uma [linguagem de programação interpretada](#). Foi originalmente implementada como parte dos [navegadores web](#) para que [scripts](#) pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade deste script passar pelo [servidor](#), controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

É atualmente a principal [linguagem para programação \*client-side\*](#) em navegadores web. Começa também a ser bastante utilizada do lado do servidor através de ambientes como o [node.js](#). Foi concebida para ser uma linguagem script com [orientação a objetos](#) baseada em protótipos, tipagem fraca e dinâmica e funções de primeira classe. Possui suporte à [programação funcional](#) e apresenta recursos como fechamentos e funções de alta ordem comumente indisponíveis em linguagens populares como [Java](#) e [C++](#). É a linguagem de programação mais utilizada do mundo.

(da Wikipédia)

### Escola online

<http://www.w3c.br/Padroes/WebDesignAplicacoes>

### Apostila Online

<https://www.caelum.com.br/apostila-html-css-javascript/>

O JavaScript é uma linguagem do lado cliente que permite interação com o visitante.

Ele tem uma linguagem parecida com o C, com o PHP e com o Java, mas com muitas diferenças. Só funciona do lado cliente, mas existe uma variante do lado server, que é o node.js.

Para escrever JavaScript no HTML tem uma certa semelhança com o CSS: nas tags, em um trecho delimitado por <script> e arquivos externos.

Exemplos:

#### 1 – No trecho com

```
<script>
function clicar(){
    alert('Clicou2');
}
</script>
```

#### 2 - Chamando na tag

Execute e veja um pequeno alert

```
<a href="#" onclick="alert('Clicou')">Clique aqui</a>
```

Chamando o script

Execute e veja um pequeno alert com Clicou2

```
<a href="#" onclick="alert(clicar())">Clique aqui</a>
```

### 3 – Em arquivo externo

Neste caso não use os comandos HTML `<script>`, use somente os comandos JS.

Exemplos:

```
<script type="text/javascript" src="/js/script.js"></script>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
```

### Exemplos de Includes

JAVASCRIPT

```
<script type="text/javascript" src="/js/script.js"></script>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
```

CSS

```
<link rel="stylesheet" type="text/css" href="mystyle.css">  
<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet">  
&@import url("//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap-glyphicons.css");
```

### Exemplos de Codificação de Caracteres

HTML

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

XML ou JavaScript ou AJAX

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

ASP:

```
<% Response.Charset="ISO-8859-1" %>
```

PHP:

```
<?php header("Content-Type: text/html; charset=UTF-8",true) ?>
```

JSP:

```
<%@ page contentType="text/html; charset=ISO-8859-1" %>
```

## 5 - Introdução ao CSS

<http://www.w3schools.com/css/default.asp>

[http://www.w3schools.com/css/tryit.asp?filename=trycss\\_ex](http://www.w3schools.com/css/tryit.asp?filename=trycss_ex)

### Escola Online

<http://www.w3c.br/Cursos/CursoCSS3>

### Ótima Apostila

<https://www.caelum.com.br/download/caelum-html-css-javascript-php.pdf>

Online

<https://www.caelum.com.br/apostila-html-css-javascript/>

**Cascading Style Sheets (CSS)** é um simples mecanismo para adicionar estilo (cores, fontes, espaçamento etc) a um documento web.

Em vez de colocar a formatação dentro do documento, o CSS cria um [link](#) (ligação) para uma página que contém os estilos. Quando quiser alterar a aparência do portal basta portanto modificar apenas um arquivo.

Com a variação de atualizações dos [navegadores \(browsers\)](#) como [Internet Explorer](#) que ficou sem nova versão de [2001](#) a [2006](#), o suporte ao CSS pode variar. O Internet Explorer 6, por exemplo, tem suporte total a CSS1 e praticamente nulo a CSS2. Navegadores mais modernos como [Google Chrome](#) e [Mozilla Firefox](#) tem suporte maior, inclusive até a CSS3, ainda em desenvolvimento.

A interpretação dos navegadores pode ser avaliada com o teste [Acid2](#), que se tornou uma forma base de revelar quão eficiente é o suporte de CSS, fazendo com que a nova versão em desenvolvimento do Firefox seja totalmente compatível a ele assim como o Opera já é. O [Doctype](#) informado ou a ausência dele determina o [quirks mode](#) ou o [strict mode](#) modificando o modo como o CSS é interpretado e a página desenhada.

(da Wikipédia)

### O que é CSS?

CSS é a abreviatura para Cascading Style Sheets. — Folha de Estilos em Cascata

### Qual é a diferença entre CSS e HTML?

HTML é usado para estruturar conteúdo. CSS é usado para formatar conteúdo estruturado.

## Sintaxe:

```
seletor {propriedade: valor;}
```

Exemplo:

```
body {color:black;}
```

ou

```
body{  
  color: black;  
}
```

Mudará a cor do texto do body para preto.

## Formas de escrever CSS em um arquivo HTML

Dentro das tags, em trechos <style> ou em arquivo externo

1 – Dentro das tags

```
<h1 style="color: #fff;">Texto em cor branca</h1>
```

2 – Em trechos com o comando HTML <style>

Geralmente entre as tags <head> e </head>

```
<head>
```

```
<style>
```

```
.titulo_vermelho{  
  color: red;  
}
```

```
#texto_azul{  
  color: blue;  
}
```

```
</style>
```

```
</head>
```

3 – Em arquivos externos. Neste caso não usamos <script> somente os comandos

Exemplos:

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

```
<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css"  
rel="stylesheet">
```



```
@import url("//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap-glyphicons.css");
```

O que mostra que quando aplicamos CSS ao HTML, o CSS mudará (sobrescreverá) o comportamento do HTML.

Existem as classes e os IDs. Veja como os criamos:

Classes

```
.nome_classe {  
    margin-left: 1.0in;  
}
```

Classe inicia com ponto (.).

ID

```
#nome_id {  
margin-left: 1.0in;  
}
```

ID inicia com cerquilha (#).

### **Veja um exemplo de layout em CSS**

Delimitando as regiões da página, tipo um template do Joomla

[http://ribafs.github.io/basicos/css/layout\\_css/index.html](http://ribafs.github.io/basicos/css/layout_css/index.html)

### **Sprites em CSS**

[http://ribafs.github.io/basicos/css/css\\_sprites.pdf](http://ribafs.github.io/basicos/css/css_sprites.pdf)

### **Template em 3 colunas do site do Moujor**

<http://ribafs.github.io/basicos/css/template3col/index.html>

### **Setas em CSS**

[http://ribafs.github.io/basicos/css/css\\_setasimples.html](http://ribafs.github.io/basicos/css/css_setasimples.html)

## 6 - Introdução ao XML

**XML** (*eXtensible Markup Language*) é uma recomendação da [W3C](#) para gerar [linguagens de marcação](#) para necessidades especiais.

É um dos subtipos da [SGML](#) (acrônimo de *Standard Generalized Markup Language* ou *Linguagem Padronizada de Marcação Genérica*) capaz de descrever diversos tipos de dados. Seu propósito principal é a facilidade de compartilhamento de informações através da [internet](#).

Entre linguagens baseadas em XML incluem-se [XHTML](#) (formato para páginas [Web](#)), [RDF](#), [SDMX](#), [SMIL](#), [MathML](#) (formato para expressões matemáticas), [NCL](#), [XBRL](#), [XSIL](#) e [SVG](#) (formato gráfico vetorial). A principal característica do XML, de criar uma infraestrutura única para diversas linguagens, é que linguagens desconhecidas e de pouco uso também podem ser definidas sem maior trabalho e sem necessidade de ser submetidas aos comitês de padronização.

(da Wikipédia)

O XML é uma linguagem de marcação muito usada como interface. O Joomla o utiliza em seus arquivos instaladores. Outros softwares fazem parecido. Já surgiu descendentes como o YML.

Exemplo de instalador de uma extensão do Joomla

### **alo.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!--?xml version="1.0" encoding="ISO-8859-1"?-->
<!DOCTYPE install SYSTEM "http://dev.joomla.org/xml/1.5/component-install.dtd">
<install type="component" version="1.5" method="upgrade">
  <name>Alo</name>
  <!-- Os elementos abaixo são opcionais -->
  <creationDate>Abril de 2009</creationDate>
  <author>Ricardo Lima Caratti</author>
  <authorEmail>ricardo@caratti.eti.br</authorEmail>
  <authorUrl>www.caratti.eti.br</authorUrl>
  <copyright>(c) 2009</copyright>
  <license>Livre</license>
  <!-- A versão do componente será registrada na tabela de componentes -->
  <version>1.0.0</version>
  <!-- A descrição é opcional -->
  <description>
    Tutorial de Construção de extensões Joomla!
    Este componente foi criado para ilustrar
    algumas técnicas de desenvolvimento de
    extensões para Joomla!
  </description>

  <!--
  As tags abaixo configuram os scripts que serão executadas na instalação
  ou desinstalação do componente (caso necessário)
```

```
<installfile>install.eventlist.php</installfile>
<uninstallfile>uninstall.eventlist.php</uninstallfile>
-->

<install>
  <sql>
    <file driver="mysql" charset="utf8">install.sql</file>
  </sql>
</install>

<uninstall>
  <sql>
    <file driver="mysql" charset="utf8">uninstall.sql</file>
  </sql>
</uninstall>

<administration>
  <menu>Alo</menu>
  <submenu>
    <menu task="show">Listar Nomes</menu>
    <menu task="add">Adicionar Nome</menu>
  </submenu>
  <files folder="admin">
    <filename>tables/alo.php</filename>
    <filename>admin.alo.php</filename>
    <filename>alo.php</filename>
    <filename>install.sql</filename>
    <filename>uninstall.sql</filename>
    <filename>admin.alo.html.php</filename>
    <filename>controller.php</filename>
    <filename>toolbar.alo.php</filename>
    <filename>toolbar.alo.html.php</filename>
  </files>
</administration>
</install>
```

Uma rápida introdução.

## 7 - Introdução ao jQuery

Aqui apenas introduções para estas ferramentas. Uma das melhores introdução é a da Wikipédia.

**jQuery** é uma [biblioteca](#) de funções [JavaScript](#) que interage com o [HTML](#) e com o CSS, desenvolvida para simplificar os [scripts](#) interpretados no [navegador](#) do cliente ([client-side](#)). Lançada em dezembro de [2006](#) no [BarCamp](#) de [Nova York](#) por John Resig. Usada por cerca de 77% dos 10 mil sites mais visitados do mundo, jQuery é a mais popular das bibliotecas JavaScript.

jQuery é uma biblioteca de [código aberto](#) e possui [licença dual](#), fazendo uso da [Licença MIT](#) ou da [GNU General Public License](#) versão 2. A sintaxe do jQuery foi desenvolvida para tornar mais simples a navegação do documento HTML, a seleção de elementos [DOM](#), criar animações, manipular eventos, desenvolver [aplicações AJAX](#) e criação de [plugins](#) sobre ela. Tais facilidades permitem aos desenvolvedores criarem camadas de [abstração](#) para interações de baixo nível de modo simplificado em aplicações web dinâmicas de grande complexidade.

A [Microsoft](#) e a [Nokia](#) anunciaram planos de incluir o jQuery em suas plataformas, a Microsoft adotou na interface de programação ([IDE](#)) [Visual Studio](#) para uso com o [framework](#) AJAX da linguagem [ASP.NET](#), e a Nokia em sua plataforma Web Run-Time de criação de pequenos aplicativos [Widgets](#). A biblioteca jQuery também tem sido usada no programa [MediaWiki](#), desde a versão 1.16

(da Wikipédia)

A jQuery hoje ganhou tanto destaque que é onipresente, tantos plugins para tudo. Se procurar plugin para algo receberá centenas, muitos.

### Exemplo

**Um código em Javascript puro**, para atribuir o valor "5" em um elemento qualquer.

```
document.getElementById( 'Teste' ).value = 5;
```

**O mesmo código em jQuery.**

```
$( '#Teste' ).val( 5 );
```

Apenas uma introdução rápida.

## 8 - Introdução do Bootstrap

O framework CSS criado pelo Twitter e oferecido free para todos é uma ferramenta que facilita a vida do programador web. Nos brinda com muita coisa que teríamos que criar manualmente, como a grid, componentes HTML/CSS diversos, como menus e outros.

Gosto muito de desenvolver aplicações (simples ou não) com o Bootstrap, que é um framework CSS que ajuda e muito no desenvolvimento front end, então criei este post para tentar apresentar o framework para quem está iniciando, principalmente o pessoal das comunidades de desenvolvimento web do Facebook, que vira e meche postam dúvidas bem básicas sobre o Bootstrap.

(<http://websocialdev.com/introducao-ao-bootstrap-para-iniciantes-em-desenvolvimento/>)

## 9 - Ambiente de Desenvolvimento

### For Windows

Para quem programa em Windows a melhor opção conheço é o Xampp

<http://xampp.sf.net>

Simple de instalar, apenas NNF (Next, Next, ... Finish).

O diretório web padrão é c:\xampp\htdocs

Para software como o Joomla ele já vem pronto.

Para alguns softwares como o CakePHP 3 requer habilitar algumas extensões, no caso mbstring (se php5), no php7 não requer. Também habilitar intl no php.ini. Depois das alterações Stop e Start o Apache.

### For Linux com PHP 7

Instalar as extensões via linha de comando.

Acessar o Terminal e executar:

```
sudo su
apt-get -y install apache2 libapache2-mod-php7.0
a2enmod rewrite;
apt-get -y install postgresql postgresql-contrib;
apt-get -y install mysql-server;
apt-get -y install php7.0 mcrypt php7.0-mcrypt php7.0-gd php7.0-mysql php7.0-pgsql php7.0-ldap;
apt-get -y install php7.0-mcrypt php-pear php7.0-xsl curl php7.0-curl phpunit php-xdebug php7.0-
intl composer;
apt-get -y install php7.0-zip php7.0-mbstring php-gettext php-mbstring php7.0-fpm php-auth
phpmyadmin;
apt-get -y install php-apcu;
wget http://ftp.ussg.iu.edu/linux/ubuntu/pool/main/m/memcached/memcached_1.4.25-
2ubuntu1_amd64.deb;
dpkg -i memcached_1.4.25-2ubuntu1_amd64.deb;
apt-get -y install php7-memcache;
```

### Habilitar o mod\_rewrite

.htaccess no Apache 2.4 trocando None por All

No /etc/apache2/apache2.conf:

```
<Directory />
    Options Indexes FollowSymLinks Includes ExecCGI
    AllowOverride All
    Order deny,allow
    Allow from all
</Directory>
```

# 10 - Introdução ao PHP

## Iniciando na Programação com PHP

Para poder programar com PHP precisamos de um servidor web. Diferente de HTML, CSS, JavaScript, XML, jQuery e Bootstrap, que nenhum destes precisa de servidor web. Podemos criar um arquivo em qualquer diretório e ele funciona.

É importante saber que atualmente o PHP deva ser a linguagem mais popular para programar para a web. Acredito que a facilidade de programar ele, os recursos oferecidos por ele, a documentação farta em seu site (<http://php.net>) e a grande aceitação da comunidade foram responsáveis pela popularidade.

Vamos mostrar na tela apenas uma frase:

```
<?php
print "Olá Mundo!";
?>
```

Se comparar isso com C, C++ ou Java vai perceber a simplicidade do PHP.

Para ver como isso funciona precisamos criar um arquivo no diretório web e chamar este arquivo no navegador. Nosso diretório web é c:\xampp\htdocs (no windows). No Linux é /var/www/html

Vamos criar neste diretório o arquivo ola.php, contendo:

```
<?php
print "Olá Mundo!";
?>
```

E chamar no navegador assim:

<http://localhost/ola.php>

Então veremos a nossa frase Olá Mundo!

## PHP com HTML

Vale lembrar que o PHP sempre fica do lado do servidor. Ele processa informações solicitadas pelo visitante e as devolve. Devolve tudo no formato HTML, juntamente com CSS e JavaScript, que ficam do lado do cliente.

Um código em PHP precisa iniciar com a TAG <?php e terminar com ?>. Qualquer arquivo que contenha código PHP precisa ter a extensão .php. Sendo assim, o interpretador do PHP não olhará para outros tipos de arquivos. E quando verificar arquivos .php somente processará os trechos entre <?php e ?>, o que estiver fora destas TAGs ele não olha.

O navegador interpreta todo código em HTML antes de mostrar na tela.

## Separador de Instruções

Observe que ao final da linha:

```
print "Como vai?";
```

Cada instrução do PHP termina com ponto e vírgula.

**Instrução** – é um comando único que damos ao PHP, no caso aqui o “print”.

Exemplo: print “Olá”;

O PHP permite escrever mais de uma instrução por linha, mas devemos evitar, por conta de que fica obscuro o código.

## Comentários

Em PHP existem três formas de se escrever comentários:

// - este é comentário oriundo do C++ e para apenas uma linha

```
/*
```

Este é o comentário oriundo do C

e para

múltiplas linhas

```
*/
```

# - Este é o comentário oriundo do shell, para uma única linha e que deve ser evitado, pois está em processo de obsolescência.

## Estruturas de Controle

Vejamos um pequeno trecho de código mais útil e interessante. Suponha que seu colégio tenha suas 4 notas dos 4 bimestres e precise fazer o cálculo para saber se você passou. Ele tem uma fórmula que é somar as 4 notas e dividir tudo por 4 para tirar a média. Veja:

Vamos criar um arquivo notas.php com o conteúdo abaixo ou testar diretamente no php-console, sem as tags inicial e final.

### IF - ELSE

```
<?php
```

```
$aluno = '<b>Elias EF</b>';
```

```
$nota1 = 10;
```

```
$nota2 = 9;
```

```
$nota3 = 7;
```

```
$nota4 = 5;
```

```
$resultado = ($nota1 + $nota2 + $nota3 + $nota4) / 4;
```

```
if( $resultado >= 7 ){
```

```
    print "O aluno $aluno foi aprovado, com nota $resultado";
```



```
}else{
    print 'O aluno não passou!';
}
?>
```

## Variáveis

Como o nome sugere seu valor varia. Vejamos.

Uma variável tem:

nome - \$aluno, que é o nome precedido de \$.

valor - 'Elias EF'

tipo – string (acontece que o PHP não declara tipos de variáveis)

## Tipos de Dados

O tipo de dados de uma variável será reconhecido quando o código for executado:

string, inteiro, float/double, boleano, array, objeto e resource.

## Nomes de Variáveis

- O nome das variáveis precisa começar com uma letra ou pelo símbolo de sublinhado
- O nome de variáveis pode conter somente caracteres alfa-numéricos e sublinhado: a-z, A-Z, 0-9 e \_ .
- Não pode conter espaços. Quando várias palavras podemos usar o sublinhado para separá-las: \$minha\_variavel.

As variáveis podem ter seu valor alterado durante a execução do script, daí seu nome.

## Atribuição

Se eu quiser guardar a primeira nota na variável \$nota1, então preciso atribuir 10 para \$nota1, assim:

```
$nota1 = 10;
```

Veja que usamos o sinal de igual para atribuir e terminamos com ponto e vírgula.

Posso atribuir um valor diretamente para uma variável, mas veja que também posso atribuir uma fórmula bem grande para a variável, assim:

```
$resultado = ($nota1 + $nota2 + $nota3 + $nota4) / 4;
```

Parecido com uma calculadora o PHP fará os cálculos e guardará o valor na variável \$resultado.

Na próxima linha ele vai testar o valor do \$resultado. (if( \$resultado >= 7 ){)

Se for maior ou igual a 7 ele mostrará a primeira mensagem.

Caso contrário então mostrará a segunda mensagem. (}else{)

Vejam os mais um exemplo prático e interessante de uso do PHP

### **IF – ELSE IF – ELSE**

```
if (expressao){
    executado se true e encerra o teste.
}else if (expressao2){
    executado se true e encerra o teste.
}else if (expressao3){
    executado se true e encerra o teste.
}else if (expressaoN){
    executado se true e encerra o teste.
}else{
    executado se false
}
```

Lembrando que somente entra no próximo teste, se o anterior for falso.

```
$hora = 14;
if($hora <= 12){
    print "Está de manhã";
}else if ($hora > 12 && $hora < 18){
    print "Está de tarde";
}else{
    print "Está de noite";
}
```

### **if - else**

O **if** testa uma expressão

Caso a expressão seja verdadeira, todo o código após o **if** será executado.

Caso não seja verdadeira o código após o **else** será executado.

## **if – eles if – else**

O **if** testa uma expressão

Caso a expressão seja verdadeira, todo o código após o **if** será executado.

Caso a expressão seja falsa então o **else if** testará uma nova expressão

Caso a nova expressão seja verdadeira o código após o **else if** será executado

Caso a nova expressão seja falsa, o código após o **else** ou o novo **else if** será executado

## **Criando Strings**

Strings pequenas, com apenas uma linha podem ser mostradas com `print` ou `echo`, mas textos com várias linhas podem ser armazenados com o recurso chamado Heredoc:

### **Armazenando o texto numa variável**

```
$str = <<<EOD
```

```
Exemplo de uma string  
distribuída em várias linhas  
utilizando a sintaxe heredoc.  
EOD;
```

### **Ou simplesmente mostrando**

```
echo <<<show
```

```
<a href="index.php">Início</a>
```

```
<h3>Edite Seus Dados</h3>
```

```
<form action="update.php" method="post">
```

```
<table width="500" border="1">
```

```
<tr>
```

```
<th scope="row">Nome</th>
```

```
<td><input type="text" name="nome" value="$nome"></td>
```

```
</tr>
```

```
<tr>
```

```
<th scope="row">Nascimento</th>
```

```
<td><input type="text" name="data_nasc" value=$data_nasc></td>
```

```
</tr>
```

```
<tr>
```

```
<th scope="row">E-mail</th>
<td><input type="text" name="email" value="$email"></td>
</tr>
</table>
</form>
</div>
show;
```

### **Mais uma estrutura interessante do PHP - Switch**

switch – Este testa uma variável contra alguns valores. Parece com um bloco de if else if. Quando encontra o valor correto executa o código seguinte e sai da iteração.

```
$opcao = 3;
switch($opcao)
{
case 1:
    print 1;
    break;
case 2:
    print 2;
    break;
case 3:
    print 3;
    break;
case 4:
    print 4;
    break;
case 5:
    print 5;
    break;
default:
    print 'Nenhuma das anteriores';
```

```
    break;
}
```

Obs.: Caso removamos um dos comandos break o código continua no próximo case.

### **Laços: for, while, forwach**

Estruturas que resolvem vários problemas com pouco esforço.

#### **for**

```
for($x = 1;$x <= 10;$x++){
    print $x;
}
```

```
for($x = 1;$x <= 10;$x++){
    print $x.'  
';
}
```

```
for($x = 0;$x < 10;$x++){
    print $x;
}
```

Existem variações de uso do for. No caso abaixo, como não temos teste do valor, o incremento continua indefinidamente:

```
for($x=0; ;$x++){
    print $x;
}
```

Nestes casos precisamos inserir um teste no corpo do for, assim:

```
for($x=0; ;$x++){
    if($x>10){
        break;
    }
    print $x;
}
```

## Continue

Este comando sai do laço e volta na próxima iteração, ou um certo número de comandos para sair do laço. Veja

```
for ($i = 0; $i < 10; ++$i) {  
    if ($i < 5){  
        continue;  
    }  
    print "$i<br>";  
}
```

## Break

Interrompe o laço, entregando o processamento para a linha seguinte ao laço.

```
for ($x = 0; $x < 5; ++$x) {  
    if ($x == 2) {  
        break;  
    }  
    print "$x<br>";  
}
```

Experimente.

## foreach

Laço através de um array.

```
$arr = array(1, 2, 3, 4);  
foreach ($arr as $value) {  
    $value = $value * 2;  
    print $value;  
}  
// $arr is now array(2, 4, 6, 8)  
unset($value); // Limpar a variável, que armazena o valor
```

## Colher Chave e Valor

```
$arr = array(1, 2, 3, 4);  
foreach ($arr as $chave => $value) {  
    $value = $value * 2;  
    print 'Chave: '.$chave.' - Valor: '.$value.'<br>';  
}  
unset($valor); // Limpar a variável, que armazena o valor
```

Existe muito mais recursos, mas me parece que já dá para perceber isso com estes exemplos mostrados.

Qualquer dúvida sobre qualquer comando do PHP vá até o site oficial em <http://php.net> e faça uma busca pelo comando, que verá a sintaxe e muitos exemplos de uso.

### **Atribuição**

Atribuir um valor para uma variável é entregar um valor para ela guardar.

Exemplo

```
$variavel = 5;
```

Estamos acima dizendo que a variável guarde o valor 5.

### **Comparação**

Comparação é o que fazemos em uma expressão geralmente em if.

Exemplo

```
if($x==3)
```

Estou querendo dizer acima: se \$x for igual a 3.

Veja que quando queremos atribuir usamos apenas um sinal de igualdade:

```
$x = 3;
```

Quando queremos comparar usamos dois sinais de igualdade: `if($x == 3)`

Com = armazenamos um valor em uma variável.

Com == eu testo se uma variável é igual a um valor.

## 11 - Introdução ao PHP Orientado a Objetos

Programação orientada a objetos é um estilo de programação que permite aos desenvolvedores agruparem tarefas semelhantes em **classes**. Isso ajuda a mantermo-nos dentro do princípio "[don't repeat yourself](#)" (**DRY**) (em português, *não repita a si mesmo*), além de facilitar a manutenção do código, pois fica dividido em módulos.

### Classes e Objetos

Classes são as matrizes dos objetos, de onde os mesmos nascem. O programador cria a classe uma única vez e depois trabalha somente com os objetos, que são instâncias das classes.

O paradigma de orientação a objetos é muito popular atualmente, sendo implementado por praticamente todas as linguagens atuais e em muitos dos softwares criados com PHP, como frameworks e CMS. Como ele diz, é orientado a objetos, simulando o mundo real com seus objetos.

Vamos tomar como exemplo um carro, que todos conhecemos e vamos modelá-lo, criando algumas propriedades e métodos para ele. As propriedades são características, como cor, já os métodos são funcionalidades como correr, frear, etc (verbos).

### Criando uma Classe

```
<?php
class Veiculo{
// Aqui fica o código da classe
}
?>
```

### Instanciando a Classe

Instanciar a classe é armazená-la em uma variável objeto para então usar seus recursos.

```
$objeto = new Veiculo(); // Também podemos usar assim: $objeto = new Veiculo;
```

As classes são compostas de propriedades e métodos. Propriedades são similares às variáveis da linguagem procedural. Métodos são similares às funções. A grande diferença é que propriedades e métodos ficam dentro da classe e todo código OO fica dentro de classes.



## Propriedades

```
<?php
class Veiculo{
    public $cor = 'Vermelha!';
}

$objeto = new Veiculo();
print 'A cor do carro é ' . $objeto->cor;
```

## 12 - Introdução ao PDO

The *PHP Data Objects* (PDO) extension defines a lightweight, consistent interface for accessing databases in PHP. Each database driver that implements the PDO interface can expose database-specific features as regular extension functions. Note that you cannot perform any database functions using the PDO extension by itself; you must use a [database-specific PDO driver](#) to access a database server.

PDO provides a *data-access* abstraction layer, which means that, regardless of which database you're using, you use the same functions to issue queries and fetch data. PDO does *not* provide a *database* abstraction; it doesn't rewrite SQL or emulate missing features. You should use a full-blown abstraction layer if you need that facility.

PDO ships with PHP 5.1, and is available as a PECL extension for PHP 5.0; PDO requires the new OO features in the core of PHP 5, and so will not run with earlier versions of PHP.

(do site oficial do PHP)

[http://php.net/manual/pt\\_BR/intro.pdo.php](http://php.net/manual/pt_BR/intro.pdo.php)

PDO é um módulo nativo do PHP que tem muito recurso para integração do PHP com SGBDs. Mas ainda assim não tem tudo. Uma certa época tive dificuldades com algumas funções e acabei por criar várias e disponibilizar no GitHub:

<https://github.com/ribafs/pdolib>

Com a chegada do PHP 7 houve uma mudança radical, não aceita mais conexão ao MySQL de forma estruturada, mas somente orientada a objetos: com o MySQLi, com o PDO ou de outra forma. Destas formas a mais popular é o PDO, pois ele dá suporte a diversos SGBDs.

Como a linguagem PHP Orientada a objetos se disseminou, o PDO também se disseminou. Veja o Zend Framework, o CakePHP, o CMS Joomla, o Drupal e muitos outros, todos orientados a objetos.

### Conexão ao MySQL

```
<?php
$dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);
?>
```

## **Conexão ao PostgreSQL**

```
<?php
$pdo = new PDO('pgsql:host=192.168.137.160;port=5432;dbname=platin', $user, $pass);
?>
```

Estas introduções são bem básicas, pois são aprofundadas nas ferramentas aprofundadas, como CakePHP 3 e Joomla 3.

## **Documentação**

[http://php.net/manual/pt\\_BR/book.pdo.php](http://php.net/manual/pt_BR/book.pdo.php)

## **Tutoriais**

<http://www.rafaelwendel.com/2011/12/tutorial-pdo-php-data-object/>

<http://www.diogomatheus.com.br/blog/php/trabalhando-com-pdo-no-php/>

<https://phpdelusions.net/pdo>

<http://www.devmedia.com.br/introducao-ao-php-pdo/24973>

## **Ferramentas**

Boa coleção de ferramentas

<https://ribafs.github.io/tools/ferramentas.pdf>

# 13 - Introdução ao MySQL

O MySQL é atualmente o SGBD free mais popular da Web.

## Teste de SQL Online

<http://www.w3schools.com/sql/default.asp>

[http://www.w3schools.com/sql/sql\\_tryit.asp](http://www.w3schools.com/sql/sql_tryit.asp)

O melhor gerenciador que considero atualmente é o Adminer, que é um gerenciador web:

<http://adminer.org>

## Administrando pela linha de comando

`mysql -h host -u root -p` (o super usuário default é root)

`mysql -u root` (sintaxe para acessar o MySQL local e quando estiver sem senha)

## Trocando a senha do Usuário root

`mysql -u root teste` (Usuário root acessar banco teste)

`use mysql;`

`UPDATE user SET Password=PASSWORD("novasenha") WHERE user="root";`

`FLUSH PRIVILEGES;`

Ou

Alterando a senha de um usuário

`mysql -uroot -psenha`

`use mysql;`

`SET PASSWORD FOR 'dnocs'@'localhost' = PASSWORD('dnocs.devel');`

## Criando Usuários com Privilégios

`mysql --user=root mysql`

`GRANT ALL PRIVILEGES ON *.* TO super@localhost`

`IDENTIFIED BY 'senha' WITH GRANT OPTION;`

`GRANT ALL PRIVILEGES ON *.* TO super@"%"`

`IDENTIFIED BY 'some_pass' WITH GRANT OPTION;`

super - é um total super usuário que pode se conectar no localhost e de qualquer lugar ("%"), mas precisa usar senha

`GRANT RELOAD,PROCESS ON *.* TO admin@localhost;`

admin - usuário que pode se conectar no localhost sem senha.

Pode executar os comandos `mysqladmin reload`, `mysqladmin refresh`, and `mysqladmin flush-*` e `mysqladmin processlist` .

Não tem nenhum privilégio relacionado aos bancos.

```
GRANT USAGE ON *.* TO fraco@localhost;
```

fraco - pode conectar somente via localhost sem senha mas sem privilégios, somente para uso.

Exemplo:

```
GRANT ALL PRIVILEGES ON *.* TO ribafs@localhost IDENTIFIED BY 'ribafs' WITH GRANT OPTION;
```

```
mysql -u ribafs // Dá erro de senha
```

```
mysql -u ribafs -p //Funciona após entrar a senha ribafs
```

### **Removendo Usuários**

```
DROP USER nomeusuario;
```

### **Privilégios**

```
REVOKE GRANT ALL ON nomebancooutabelaou*ou*.* FROM nomeusuario
```

\* - todas as tabelas

\*.\* todos os bancos e todas as tabelas

banco.\* - todas as tabelas do banco

```
GRANT SELECT,INSERT,UPDATE ON nomebanco.* TO nomeuser;
```

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP ON nomebanco.* TO usuario@localhost IDENTIFIED BY 'senha';
```

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP ON nomebanco.* TO usuario@dominio.com.br IDENTIFIED BY 'senha';
```

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP ON nomebanco.* TO usuario@%' IDENTIFIED BY 'senha';
```

### **Remover um serviço:**

```
bin\mysqld --remove mysql
```

### **Criar Tabelas Relacionadas**

```
create table produto(  
    codigo int not null primary key,  
    nome varchar(50) not null unique,  
    descricao varchar(200),  
    valor real(6, 2)  
) ENGINE=INNODB;
```

```
create table cliente(  
    codigo int not null primary key,  
    nome varchar(50) not null,
```

```
    email varchar(100) not null unique,  
    cpf varchar(11) not null  
) ENGINE=INNODB;
```

```
create table pedido(  
    numero int not null primary key auto_increment,  
    codigocliente int not null references cliente(codigo),  
    valortotal real(7,2) DEFAULT '0.00' NOT NULL  
) ENGINE=INNODB;
```

```
create table item(  
    numeropedido int not null references pedido(numero),  
    codigoproduto int not null references produto(codigo),  
    quantidade int not null,  
    primary key(numeropedido, codigoproduto)  
) ENGINE=INNODB;
```

```
CREATE TABLE product (  
    category INT NOT NULL,  
    id INT NOT NULL,  
    price DECIMAL,  
    PRIMARY KEY(category, id)  
) ENGINE=INNODB;
```

```
CREATE TABLE product_order (  
    no INT NOT NULL AUTO_INCREMENT,  
    product_category INT NOT NULL,  
    product_id INT NOT NULL,  
    customer_id INT NOT NULL,  
    PRIMARY KEY(no),  
    INDEX (product_category, product_id),  
    FOREIGN KEY (product_category, product_id)  
    REFERENCES product(category, id) ON UPDATE CASCADE ON DELETE RESTRICT,  
    INDEX (customer_id),  
    FOREIGN KEY (customer_id)  
    REFERENCES customer(id)  
) ENGINE=INNODB;
```

O tipo InnoDB dá suporte à constraint Foreign Key (references).

## **Importar e Exportar**

### ***Exportando:***

```
bin\mysqldump -u user -p passwd banco > banco.sql
```

### ***Importando:***

```
bin\mysql -u user -p password banco < banco.sql
```

Mudar Conjunto de Caracters para LATIN1  
mysql -u root  
\C latin1

### **Funções com Datas**

DATE\_SUB  
SELECT something FROM tbl\_name WHERE DATE\_SUB(CURDATE(),INTERVAL 30 DAY) <=  
date\_col;  
SELECT DATEDIFF('1997-12-31 23:59:59','1997-12-30');

DATE\_ADD  
SELECT DATE\_ADD('2006-05-00',INTERVAL 1 DAY);

SELECT CURDATE();

SELECT CURTIME();

DATE\_FORMAT  
SELECT date\_format( '2006-04-30', '%d/%m/%Y' ); -- 30/04/2006  
SELECT DATE\_FORMAT('2003-10-03',GET\_FORMAT(DATE,'EUR')); -- 03.10.2003  
SELECT DATE\_FORMAT('2006-06-00', '%d/%m/%Y');

SELECT NOW();

SELECT TO\_DAYS('1997-10-07'); -- RETORNA DIAS

SELECT YEAR('2000-01-01');

### **Metadados do MySQL**

Consulta com metadados para retornar os nomes de todas as tabelas de um banco:

```
SELECT * FROM information_schema.tables WHERE table_type = 'BASE TABLE' AND  
table_schema = 'joomla';
```

```
select * from information_schema.tables where table_schema='condominio' and  
table_name='despesas';
```

```
SELECT column_name, column_type FROM INFORMATION_SCHEMA.columns where  
table_name='despesas'
```

### **Field Type**

```
SELECT column_name, column_type FROM INFORMATION_SCHEMA.columns where  
table_name='despesas' and column_name='descricao'
```

Ou

```
SHOW COLUMNS FROM despesas WHERE Field = 'descricao'
```

### **Field Len**

```
SELECT character_maximum_length as len FROM INFORMATION_SCHEMA.columns where  
table_name='$table' and column_name='$fldName'
```

```
SELECT VERSION()      Server version string  
SELECT DATABASE()    Current database name (empty if none)  
SELECT USER()        Current username  
SHOW STATUS          Server status indicators  
SHOW VARIABLES       Server configuration variables
```

Consulta com metadados para retornar os nomes de todas as tabelas de um banco:

```
SELECT * FROM information_schema.tables WHERE table_type = 'BASE TABLE' AND  
table_schema = 'joomla';
```

### **Retornar todos os bancos:**

```
SELECT * FROM INFORMATION_SCHEMA.schemata
```

```
SELECT TABLE_SCHEMA, TABLE_NAME, ENGINE, TABLE_ROWS, DATA_LENGTH,  
AUTO_INCREMENT FROM TABLES WHERE TABLE_SCHEMA = 'nomeBanco';
```

```
SELECT TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME, COLUMN_TYPE,  
COLLATION_NAME FROM COLUMNS WHERE 1 = 1 AND TABLE_SCHEMA='condominio'  
AND COLUMN_NAME='descricao';
```

```
SELECT COLLATION('abc')
```

```
SHOW TABLES
```

### **Primary Key de uma tabela**

```
SHOW INDEX FROM despesas
```

Foreign Key (somente para tabela com chave estrangeira)

```
select column_name, concat(referenced_table_name, '.', referenced_column_name) as 'references'  
from information_schema.key_column_usage where referenced_table_name is not null and  
table_name='$table';
```

-- Retorna todas as chaves estrangeiras

```
use information_schema;  
show tables;
```

### **Transações no MySQL**

Iniciar uma transação:

```
set autocommit=0;  
start transaction; -- Para versões anteriores a 4 usa-se, ao invés, begin;
```

```
insert into conteudo values (21, 'teste','Teste', 'TESTE');
```

```
commit; -- Caso queira confirmar  
rollback; -- Caso queira cancelar os comandos
```



# 14 - CMS Joomla

## 14.1 - Instalação do CMS Joomla 3.6.5

O Joomla é um dos melhores e mais populares CMS existentes atualmente para a criação de sites, tipo portal.

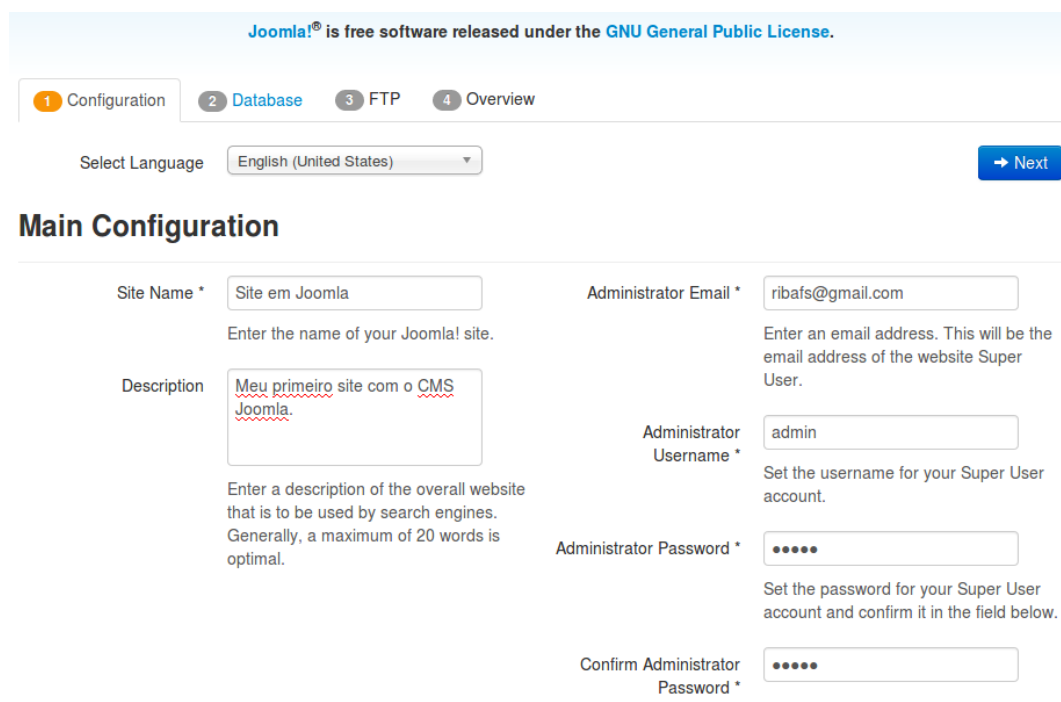
### Download

<https://www.joomla.org/>

Para instalar localmente em seu desktop não requer criação do banco sendo MySQL, apenas use o super usuário root. Se num servidor crie antes o banco.

Após efetuar o download apenas crie uma pasta no document root e descompacte o arquivo baixado dentro dela.

Arqui vou criar uma pasta chamada **portal**.



The screenshot shows the Joomla! installation configuration interface. At the top, it states "Joomla!® is free software released under the GNU General Public License." Below this, there are four tabs: "1 Configuration" (active), "2 Database", "3 FTP", and "4 Overview". A "Select Language" dropdown is set to "English (United States)", and a "Next" button is visible. The "Main Configuration" section contains several fields:

- Site Name \***: Input field with "Site em Joomla". Below it: "Enter the name of your Joomla! site."
- Description**: Textarea with "Meu primeiro site com o CMS Joomla.". Below it: "Enter a description of the overall website that is to be used by search engines. Generally, a maximum of 20 words is optimal."
- Administrator Email \***: Input field with "ribafs@gmail.com". Below it: "Enter an email address. This will be the email address of the website Super User."
- Administrator Username \***: Input field with "admin". Below it: "Set the username for your Super User account."
- Administrator Password \***: Password field with masked characters. Below it: "Set the password for your Super User account and confirm it in the field below."
- Confirm Administrator Password \***: Password field with masked characters.

Na sequência, veja os valores de exemplo acima:

Escolha o idioma – Português do Brasil

Nome do Site

Descrição

Email do Administrador

Nome do Administrador

Senha do Administrador

Confirmar Senha do Administrador

Localmente gosto de usar admin e admin para user e senha. Próximo

1 Configuração 2 Banco de dados 3 FTP 4 Visão geral

### Configuração do Banco de Dados

← Anterior → Próximo

Tipo de Banco de dados \*  Provavelmente é "MySQL".

Nome do servidor \*  Costuma ser "localhost".

Nome de Usuário \*  Algo como "root" ou um nome de usuário dado pelo local de hospedagem.

Senha  Para a segurança do site, o uso de uma senha para a conta do banco de dados é obrigatório.

Nome do banco de dados \*  Alguns servidores permitem apenas um nome específico de Banco de Dados por site. Neste caso, use o prefixo de tabelas para diferentes sites Joomla.

Prefixo das tabelas \*  Crie um prefixo de tabela ou use o prefixo gerado aleatoriamente. **O ideal é que tenha quatro ou cinco caracteres, contenha apenas caracteres alfanuméricos e TERMINE com um sublinhado. Certifique-se que**

Dados:

Nome do servidor

Nome do usuário

Senha

Banco de dados

Depois de preencher clique em Próximo

Na tela de Configuração do FTP apenas clique em Próximo

 Joomla!®

Joomla!® é um software gratuito distribuído sob a Licença GNU/GPL.

1 Configuração 2 Banco de dados 3 FTP 4 Visão geral

### Finalização

← Anterior → Instalar

Instalar conteúdo de exemplo  Nenhum (Obrigatório para a criação básica de site multi-idiomas)

Conteúdo de exemplo Blog, em inglês (GB)

Conteúdo de exemplo Boletim, em inglês (GB)

Conteúdo de exemplo padrão, em inglês (GB)

Conteúdo de exemplo Aprenda Joomla, em inglês (GB)

Instalar o conteúdo de exemplo é altamente recomendável para iniciantes. Será instalado o conteúdo de exemplo que está incluído no pacote de instalação Joomla.

### Visão geral

Configuração de email

Enviar as configurações por email para [ribafs@gmail.com](mailto:ribafs@gmail.com) após a instalação.

Para finalizar mantenha Instalar conteúdo de exemplo – Nenhum e clique em Instalar.

Veja o que aparece

**Parabéns! O Joomla foi instalado com sucesso.**

### Joomla! em seu idioma e/ou criação básica de um site multi-idioma

Antes de remover o diretório installation, você pode instalar outros idiomas. Se deseja adicionar outros idiomas em seu aplicativo Joomla!, clique no botão a seguir.

[→ Etapas extras: instalar idiomas](#)

Obs.: você precisará de acesso à internet para que o Joomla! baixe e instale novas idiomas. Algumas configurações de servidor não vão permitir que o Joomla! instale os idiomas. Se este for seu caso, não se preocupe, você poderá instalá-los depois pelo Administrador.

LEMBRE DE REMOVER O DIRETÓRIO installation.  
Não será possível prosseguir até que o diretório installation seja removido. Este é um recurso de segurança do Joomla!

[Remover o diretório installation](#)

[Site](#) [Administrador](#)

### Detalhes de Acesso ao

Caso queira instalar um site com multi-idioma clique no Etapas extras: instalar idioma. Como não usarei este recurso apenas clique em

### Remover o diretório Installation

E depois clique em **Administrador**

Aparece a tela de login



Assim teremos um site limpo, sem conteúdo de exemplo, para mais facilmente montar um site em produção.

### Sugestão

Para a primeira instalação sem orientação prefira uma das instalações com conteúdo de exemplo.

No Windows não terá dificuldades.

No Linux, caso não gerencie bem as permissões, precisará criar o configuration.php e colar nele o conteúdo sugerido e também remover o diretório installation.

## 14.2 - Gerenciando Template para Joomla com Gantry

### Download

<http://gantry.org/downloads>

Role a tela para selecionar Joomla

Faça o download do Component ([joomla-pkg\\_gantry5\\_v5.4.11.zip](#)) e do template Hydrogen ([joomla-tpl\\_g5\\_hydrogen\\_v5.4.11.zip](#)).

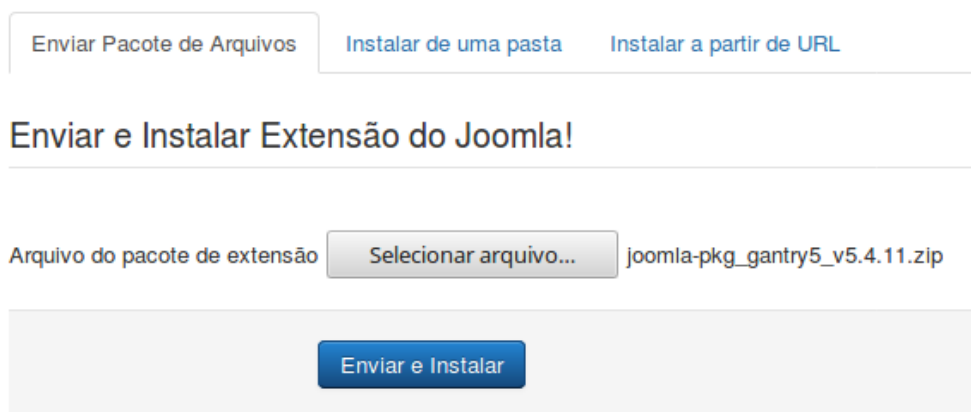
### Instalação

#### Componente

Acesse a seção de administração

Clique em Extensões – Gerenciar – Instalar

Selecione o “joomla-pkg\_gantry5\_v5.4.11.zip” do componente, como indicado abaixo:



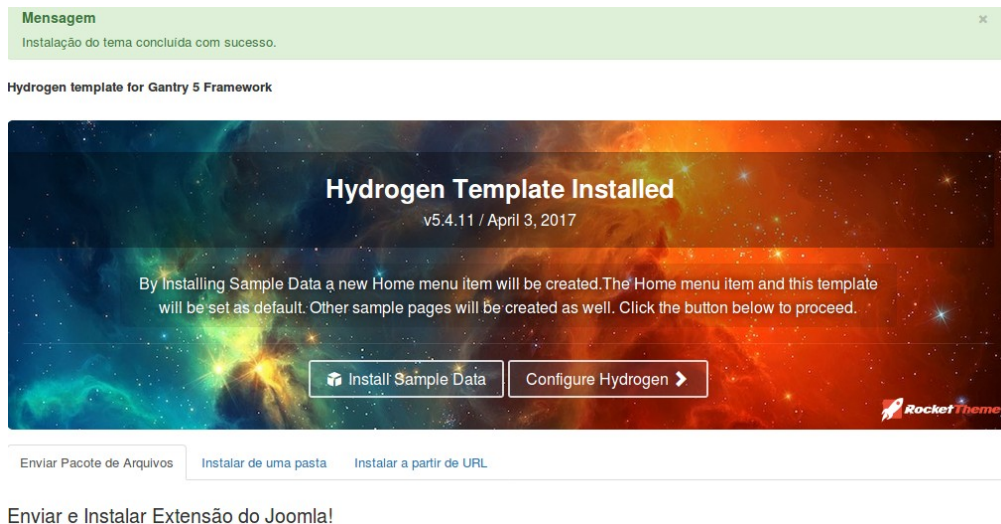
The screenshot shows the Joomla! extension installation interface. At the top, there are three tabs: "Enviar Pacote de Arquivos" (selected), "Instalar de uma pasta", and "Instalar a partir de URL". Below the tabs is the heading "Enviar e Instalar Extensão do Joomla!". Underneath, there is a label "Arquivo do pacote de extensão" followed by a "Selecionar arquivo..." button and the filename "joomla-pkg\_gantry5\_v5.4.11.zip". At the bottom, there is a large blue button labeled "Enviar e Instalar".

E clique em Enviar e Instalar.

## Instalar o Template

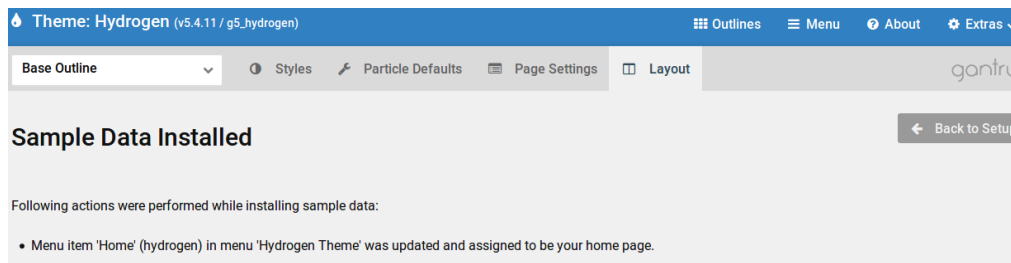
Proceder de forma semelhante à instalação do componente e instale o “joomla-tpl\_g5\_hydrogen\_v5.4.11.zip”.

Aparece a tela



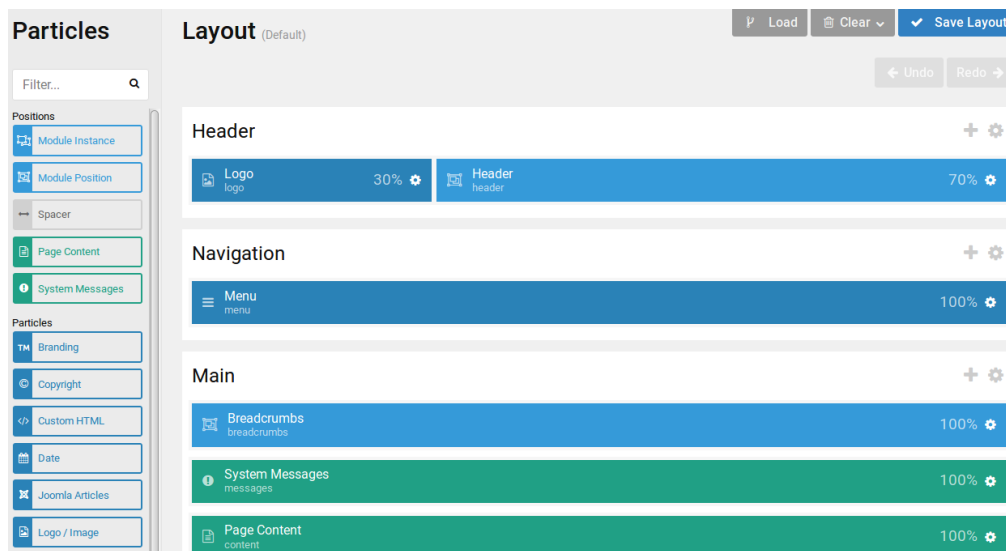
Então clique em  
Install Sample Data

Então recebemos a tela



De todos os muitos recursos existentes basicamente usaremos o Layout

## Clique em Layout



Da esquerda basicamente usaremos Module Instance para inserir módulos existentes. Funciona assim:

- Instalamos e habilitamos um módulo pelo Joomla
- Vimos ao Gantry e o instalamos em uma posição do Layout

Existem prontas as posições:

Header

Logo Header

Navigation

Menu

Main

Breadcrumbs

System Message

Page Content

Footer

Footer

Copyright

Spacer

Branding

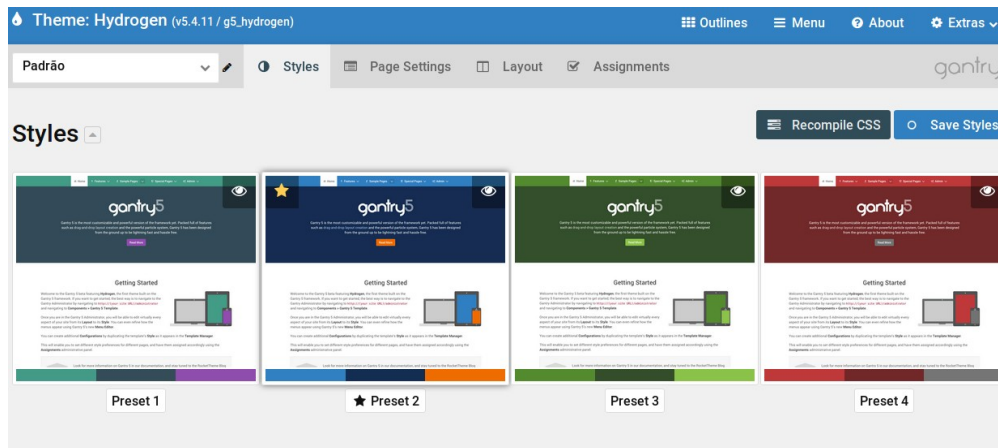
Offcanvas Section

Mobili-menu

## Configurações

Extensões – Temas – (já está marcado como padrão o Hydrogen Padrão). Clique nele.

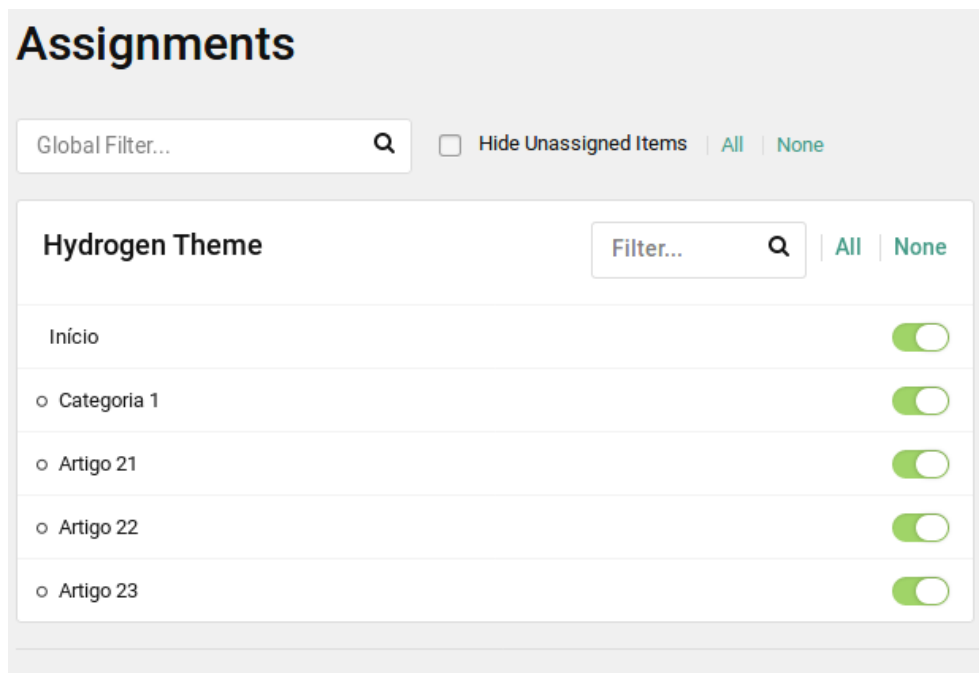
Escolher o estilo a ser adotado clicando num dos 4 Presets



E clique em Save Styles

**Atribuir Menus**

Clique acima em Assignments



## Usando o painel Layout do Gantry

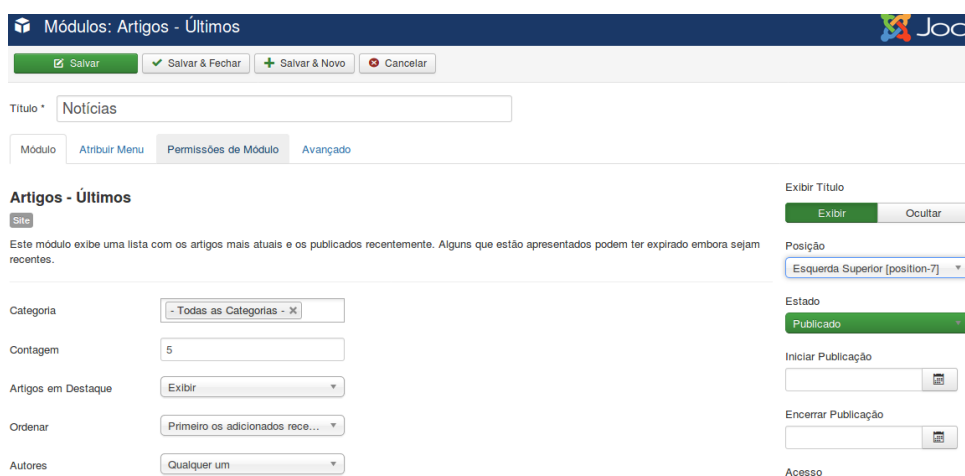
O Joomla tem dois módulos nativos que acredito valorizam sites, que é o Novidades e o Populares.

Vamos adicionar ambos na seção de módulos do Joomla.

## Adicionando o site Novidades

Extensões – Módulos – Novo

O que chamei de Novidades está na relação do Joomla como Artigos – Últimas Notícias



The screenshot shows the Joomla! administrator interface for adding a new module. The title is 'Notícias' and the position is 'Esquerda Superior [position-7]'. The state is set to 'Publicado'. The interface includes a search bar, a list of modules, and a form for adding a new module. The form fields include: Categoria (Todas as Categorias - X), Contagem (5), Artigos em Destaque (Exibir), Ordenar (Primeiro os adicionados rece...), Autores (Qualquer um), and Estado (Publicado). There are also buttons for 'Exibir Título', 'Exibir', and 'Ocultar'.

Entrei apenas com o Título e a Posição, position-7

De forma semelhante criar o Populares, que na relação do Joomla é o Artigos – Mais Lidos, na posição position-8.

Se procurar no sites agora elas não aparecerão, pois o Gantry controla o layout. Teremos que adicionar ao Layout do template Hydrogen.

## Adicionando Módulo ao Layout do Hydrogen

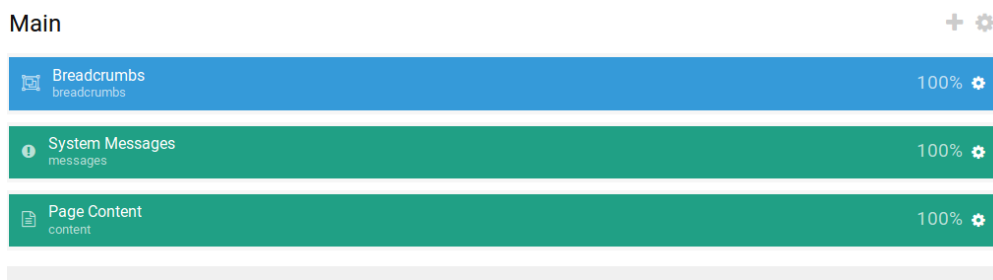
Extensões – Temas – Hidrogen Padrão  
Layout

Clique acima e à esquerda em Padrão e selecione Base Outline para que o Layout seja desbloqueado.

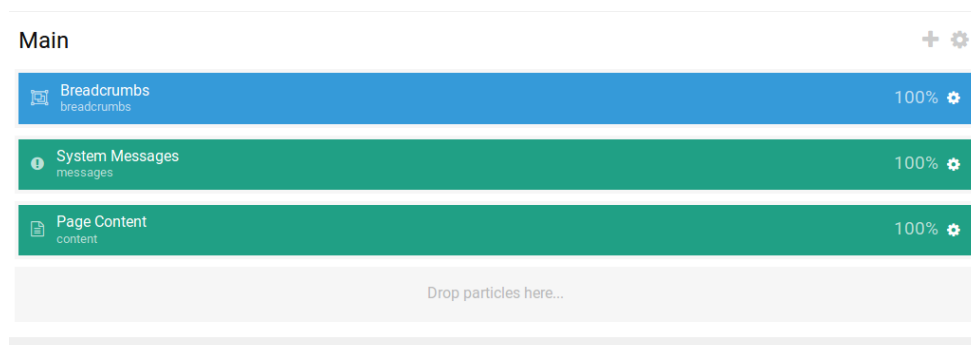
Agora é o momento de selecionar uma área do Layout para adicionar os dois módulos.



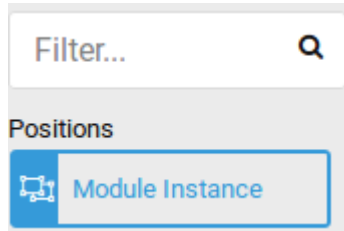
Vamos criar uma linha abaixo de Page Content, da seção Main



Para isso clicamos uma vez no sinal de mais acima e à direita. Isso inserirá uma linha em branco dessa região, veja:

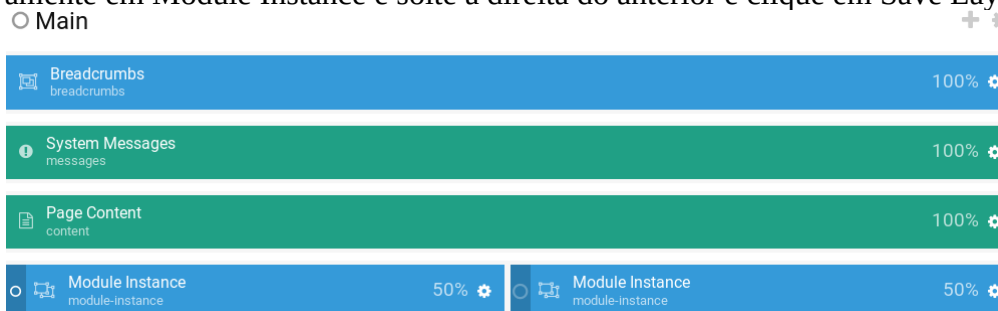


Na linha abaixo (Drop particles here...) adicionaremos dois Module Instance. Veja da esquerda acima.



Clicar em Module Instance e arrastar soltando em Drop particle here.

Clique novamente em Module Instance e solte à direita do anterior e clique em Save Layout. Veja



## Configurando as Posições dos módulos

Clique na pequena roldana do Module Instance da esquerda.

Quando clica abre-se

The screenshot shows the Joomla! Particle configuration interface. At the top, there are tabs for 'Particle', 'Block', and 'Inheritance'. The 'Particle' tab is active. Below the tabs, the module name 'Novidades' is displayed with a pencil icon and a 'module' label. A green toggle switch is turned on. A blue banner contains the text: 'To edit the Module please use the Joomla's Module Manager' with an external link icon. Below this, there is a 'Module Id' field containing the value '88' and a 'Pick a Module' button. Underneath, there is a 'Chrome' field containing the value 'gantry' and a checkbox. At the bottom of the window, there are three buttons: 'Apply', 'Apply and Save', and 'Cancel'.

Clique no pequeno lápis acima e renomeie para Novidades. Então clique no botão Pick a Module.

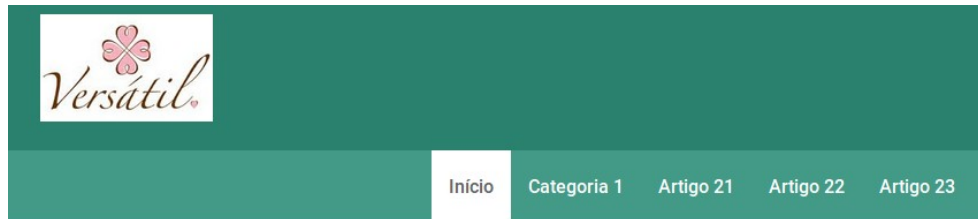
Abre-se

The screenshot shows the 'Pick a Module' dialog box. It has a title bar 'Pick a Module' and a search bar with the placeholder text 'Search...'. Below the search bar, there is a grid of module cards. Each card displays the module name, ID, Type, and Position. The 'Notícias' module (ID: 88, Type: mod\_articles\_latest, Position: position-7) is highlighted with a black border. Other visible modules include 'Gantry 5 Particle' (ID: 87), 'Breadcrumbs' (ID: 17), 'Login Form' (ID: 16), 'Main Menu' (ID: 1), and 'Populares' (ID: 89). At the bottom right of the dialog, there are 'Select' and 'Cancel' buttons.

Nesta janela clique no módulo Notícias e então clique em Select. Volta para a janela anterior, então click em Apply and Save.

Faça semelhante com o módulo Populares.

Agora veja o site com nossos novos módulos.



### Notícias

- [Artigo 11](#)
- [Artigo 12](#)
- [Artigo 13](#)
- [Artigo 21](#)
- [Artigo 22](#)

### Populares

- [Artigo 11](#)
- [Artigo 12](#)
- [Artigo 13](#)
- [Artigo 21](#)
- [Artigo 22](#)

## Usando a Marca original do Gantry

Podemos alterar a marca registrada que aparece no site:

“Powered by [Gantry Framework](#)“

Em Particles Default – Branding

Tanto podemos alterar como também desabilitar clicando no botão.

## Posições do Hydrogen

Usando a região Header do gantry.

O gantry já traz algumas posições (header, debug, breadcrumbs e footer). Se adicionarmos módulos nestas posições eles já aparecem sem usar o Layout.

Vamos usar a header:

- Crie um novo módulo tipo HTML Personalizado.
- Mude seu título para Header
- Oculte seu Título
- Insira uma imagem

Salve e Feche

Veja no site o resultado.

## Outra boa dica

Mudar o Módulo existente para a posição breadcrumbs do Gantry. Uma beleza.

Veja no site.

Após as alterações acima agora o painel Page Settings agora mostra uma partícula para Header. Podemos inserir conteúdo aí que imediatamente aparece no topo do site.

### Um bom template que usa Gantry



Você está aqui: Página Principal

#### Notícias

- [Artigo 11](#)
- [Artigo 12](#)
- [Artigo 13](#)
- [Artigo 21](#)
- [Artigo 22](#)

#### Populares

- [Artigo 11](#)
- [Artigo 12](#)
- [Artigo 13](#)
- [Artigo 21](#)
- [Artigo 22](#)

[Joomla 3x Templates/Joomla 3x Templates/JL Learnmate Free.html](http://www.joomla24.com/)

## 14.3 - Gerenciamento de Conteúdo do Joomla 3.6.5

Neste tutorial iremos cuidar de

- categorias
- artigos
- itens de menu

Meu projeto aqui é o seguinte:

### Layout do Conteúdo

Duas Categorias:

- Categoria 1
- Categoria 2

Seis artigos

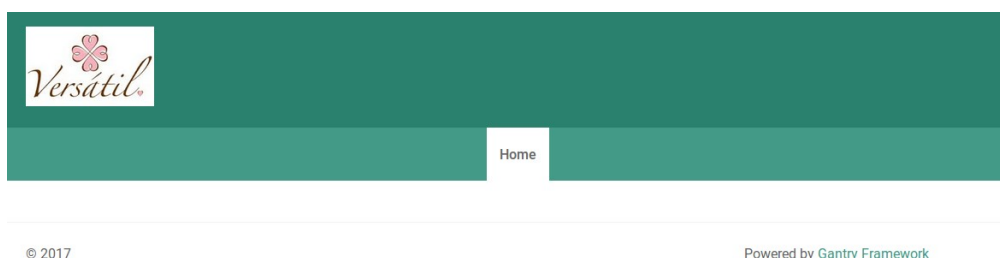
- Categoria 1
  - Artigo 11
  - Artigo 12
  - Artigo 13
- Categoria 2
  - Artigo 21
  - Artigo 22
  - Artigo 23

Quatro itens de menu, sendo 1 chamado Categoria 1 para a Categoria 1  
Sendo 3 abrindo cada um dos 3 artigos da categoria 2

Os números dos artigos identificam o artigo na categoria. Exemplo:  
13 – categoria 1 e artigo 3

Iniciaremos com a tela limpa, sem conteúdo (foi a nossa escolha ao instalar). Iremos somente criar conteúdo válido (apenas como se fosse um site válido). Se instalarmos conteúdo de exemplo, usaremos o mesmo somente para testes, aprendizado ou teremos que remover todo o conteúdo de exemplo e criar o nosso válido.

Veja o site:

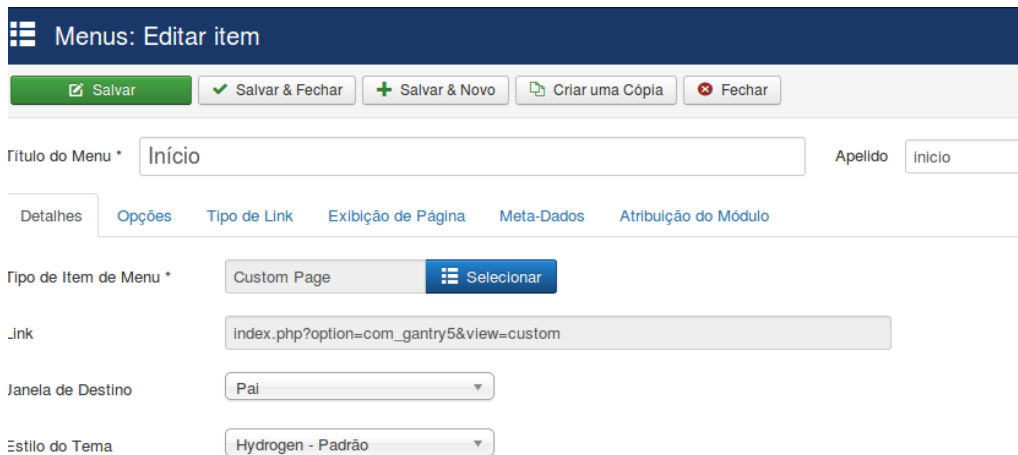


Primeira providência: mudar de Home para Início, inclusive o apelido para “início”.

## Alterar Item de Menu Home

Menus – Hydrogen Theme

Clique em Home e troque o Título do menu e o Apelido assim:



Menus: Editar item

Salvar Salvar & Fechar Salvar & Novo Criar uma Cópia Fechar

Título do Menu \* Início Apelido inicio

Detalhes Opções Tipo de Link Exibição de Página Meta-Dados Atribuição do Módulo

Tipo de Item de Menu \* Custom Page Selecionar

Link index.php?option=com\_gantry5&view=custom

Janela de Destino Pai

Estilo do Tema Hydrogen - Padrão

E clique em Salvar e Fechar

Vá novamente ao site, na frente, e reinicie o navegador para ver a alteração.

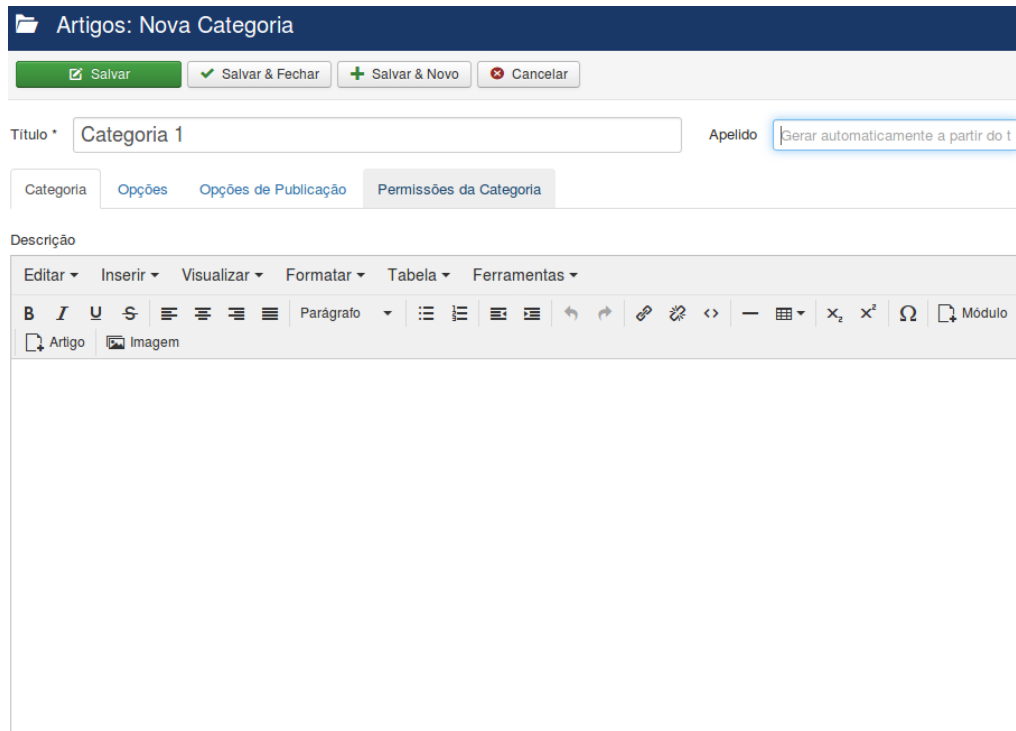
## Criação de Categorias

Na criação de categorias apenas o Título é requerido, então as criaremos com apenas o título.

Conteúdo – Categorias – Criar Nova Categoria

## Criar a Categoria 1

De forma semelhante criar a Categoria 2.



The screenshot shows a web interface for creating a new category. At the top, there is a dark blue header with the text "Artigos: Nova Categoria". Below the header, there is a row of buttons: "Salvar" (green), "Salvar & Fechar" (green with checkmark), "Salvar & Novo" (green with plus), and "Cancelar" (grey with red X). Below the buttons, there are two input fields: "Titulo" with the value "Categoria 1" and "Apelido" with the value "Gerar automaticamente a partir do t". Below the input fields, there are four tabs: "Categoria" (selected), "Opções", "Opções de Publicação", and "Permissões da Categoria". Below the tabs, there is a section labeled "Descrição" with a rich text editor. The editor has a menu with "Editar", "Inserir", "Visualizar", "Formatar", "Tabela", and "Ferramentas". The toolbar includes icons for bold, italic, underline, strikethrough, bulleted list, numbered list, indent, outdent, paragraph, link, unlink, undo, redo, source code, zoom in, zoom out, and insert link. There are also buttons for "Artigo" and "Imagem".

## Criação de Artigos

Já os artigos precisam de:

- Título
- Conteúdo
- Categoria

Iremos criar 3 na categoria 1 e 3 na categoria 2, como mostrado acima.

Criarei apenas um artigo, com detalhes e você cria todos os 6 de forma semelhante.

## Criar o Artigo 11

Título – Artigo 11

Categoria – Categoria 1

Conteúdo – veja a imagem

The screenshot shows the 'Artigos: Editar' (Articles: Edit) interface. At the top, there are buttons for 'Salvar', 'Salvar & Fechar', 'Salvar & Novo', 'Criar uma Cópia', 'Versões', and 'Fechar'. Below these, the title 'Artigo 11' and the alias 'artigo-11' are visible. The main content area shows the article's HTML definition and examples. On the right side, there are settings for 'Estado' (Publicado), 'Categoria' (Categoria 1), 'Destaque' (Sim), 'Acesso' (Public), 'Idioma' (Todos), 'Marcadores', and 'Observação da \'

Veja os 6 artigos abaixo

The screenshot shows a search results page with a green message box at the top stating 'Mensagem: Artigo salvo com sucesso'. Below the message is a search bar with the text 'Pesquisar' and a search icon. To the right of the search bar are buttons for 'Ferramentas de Busca' and 'Limpar'. The search results are displayed in a table with columns for 'Estado', 'Título', and 'Nível'. The table contains 6 rows of article information.

Estado	Título	Nível
<input checked="" type="checkbox"/>	Artigo 23 (Apelido: artigo-23) Categoria: Categoria 2	Public
<input checked="" type="checkbox"/>	Artigo 22 (Apelido: artigo-22) Categoria: Categoria 2	Public
<input checked="" type="checkbox"/>	Artigo 21 (Apelido: artigo-14) Categoria: Categoria 2	Public
<input checked="" type="checkbox"/>	Artigo 13 (Apelido: artigo-13) Categoria: Categoria 1	Public
<input checked="" type="checkbox"/>	Artigo 12 (Apelido: artigo-12) Categoria: Categoria 1	Public
<input checked="" type="checkbox"/>	Artigo 11 (Apelido: artigo-11) Categoria: Categoria 1	Public



## Vamos agora Criar os Itens de Menu

Como adotamos o template Hydrogen do Gantry, iremos usar o menu Hydrogen Theme

Criaremos 3 itens de menu, um para cada artigo da categoria 2.  
Mais 1 item para abrir a categoria 1.

### Criação de Item de Menu

Menus - Hydrogen Theme – Adicionar Item de Menu

Primeira Etapa

Menus: Novo item

Salvar Salvar & Fechar Salvar & Novo Cancelar

Titulo do Menu \* Categoria 1 Apelido Gerar auto

Detalhes Tipo de Link Exibição de Página Meta-Dados Atribuição do Módulo

Tipo de Item de Menu \* Selecionar

Link

Janela de Destino Pai

Estilo do Tema - Usar Padrão -

Na segunda etapa clicamos em Selecionar para escolher a Categoria 1 assim

Para realizar esta etapa:

- Clique em Selecionar
- Clique acima em Artigos
- Lista de uma Categoria

Selecione uma categoria – Categoria 1

Salvar e  
Fechar

### Menus: Novo item

Título do Menu \*

[Detalhes](#) [Categoria](#) [Layouts de Lista](#) [Opções](#) [Opções da integração](#) [Tip](#)

Tipo de Item de Menu \*

Selecione uma categoria \*

Link

Janela de Destino

Estilo do Tema

## Criar agora os 3 Itens de Menu, sendo um para cada artigo.

- Novo Item de Menu

Menus - Hydrogen Theme – Adicionar Item de Menu

Título do Menu – Artigo 21

Selecionar – Artigos

Artigo Especificado

Selecione um artigo – Artigo 21 Selecionar

Obs.: o Apelido será adicionado automaticamente (você pode criar um customizado).

Menus: Novo item

Salvar Salvar & Fechar Salvar & Novo Cancelar

Título do Menu \* Artigo 21 Apelido Gerar automático

Detalhes Opções Tipo de Link Exibição de Página Meta-Dados Atribuição do Módulo

Tipo de Item de Menu \* Artigo Especifico Selecionar

Selecione um Artigo \* Artigo 21 Selecionar

Link index.php?option=com\_content&view=article

Menu de Destino Pai

Estilo do Tema - Usar Padrão -

Salvar e Fechar

Crie os outros 2 restantes de forma semelhante.

Veja o resultado do nosso trabalho visualizando a frente do site no navegador.

## 14.4 - Administração do Joomla 3.6.5

Aqui abordarei a instalação de extensões de terceiros.

Já abordamos a criação de conteúdo e o gerenciamento de templates, usando o Gantry, agora vamos abordar o gerenciamento de extensões de terceiros.

### Instalação do Componente Simple Backup

Este componente realiza um backup full de um site com Joomla. O backup gerado é feito com a geração de dois pacotes/arquivos, um contendo todos os arquivos e o outro com o banco de dados.

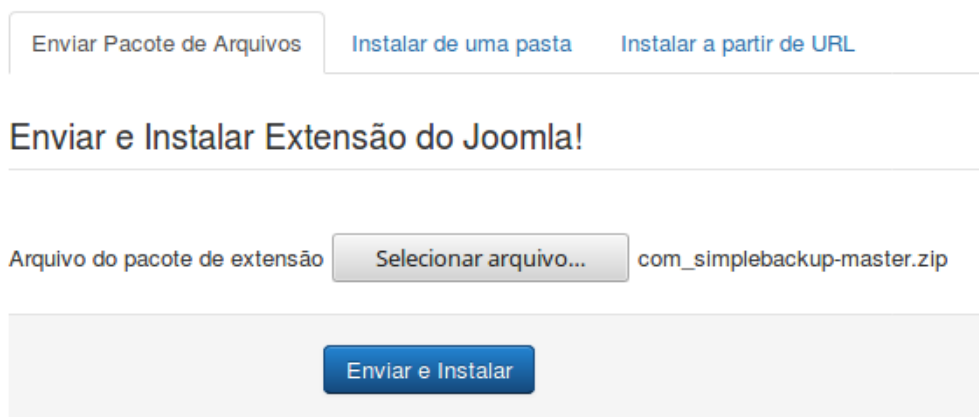
Com apenas 3 cliques o backup é realizado de forma bem simples, fazendo jus ao nome.

### Download

[https://github.com/ribafs/com\\_simplebackup](https://github.com/ribafs/com_simplebackup)

### Instalação

Extensões – Gerenciar – Instalar



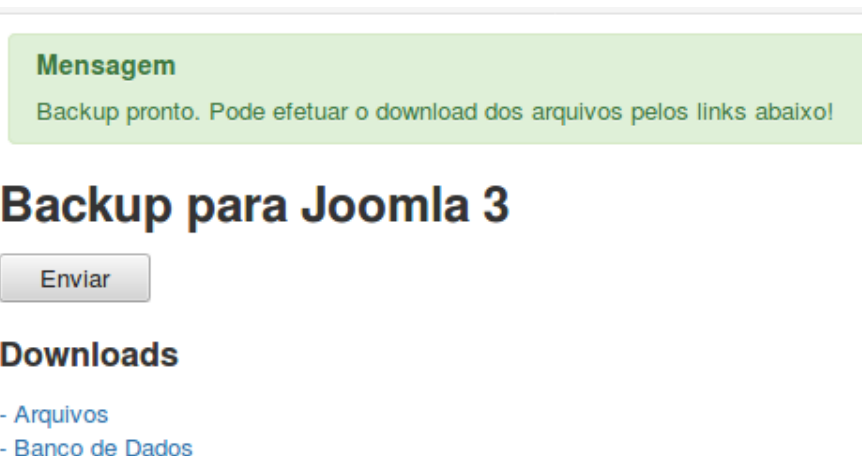
The screenshot shows the Joomla! administration interface for installing an extension. At the top, there are three tabs: "Enviar Pacote de Arquivos" (selected), "Instalar de uma pasta", and "Instalar a partir de URL". Below the tabs, the heading "Enviar e Instalar Extensão do Joomla!" is displayed. Underneath, there is a label "Arquivo do pacote de extensão" followed by a file selection button labeled "Selecionar arquivo..." and the filename "com\_simplebackup-master.zip". At the bottom of the form, there is a prominent blue button labeled "Enviar e Instalar".

Veja que selecionei o arquivo.

Agora clique em Enviar e Instalar

## Vejamos sua interface

Componentes – Backup Simples



The screenshot shows a Joomla! interface for a backup component. At the top, there is a green message box with the text: "Mensagem Backup pronto. Pode efetuar o download dos arquivos pelos links abaixo!". Below this is the title "Backup para Joomla 3" in a large, bold font. Underneath the title is a button labeled "Enviar". Below the button is the section header "Downloads" followed by two blue links: "- Arquivos" and "- Banco de Dados".

- Primeiro clique em Enviara
- Aguarde a geração do backup
- Faça o download clicando em Arquivos e em Banco de Dados

Guardar e usar somente em caso de desastre do site atual.

É uma boa ideia efetuar backup com certa frequência e guardar bem guardado.

Caso precise restaurar e não saiba o que fazer veja:

<https://ribafs.github.io/joomla/programacao/restore.html>

Caso o domínio já tenha sido propagado estará aqui:

<http://ribafs.org/joomla/programacao/restore.html>

## Renomear Usuário

**Dica:** ao criar um novo site é bom criar outro usuário ou renomear o existente para seu nome.

Usuários – Gerenciar

Clique no nome e altere.

## Módulo Polivalente

O Joomla tem um módulo chamado HTML Personalizado

Este módulo tem basicamente uma textarea que pode abrigar qualquer dos componentes suportados pelo Joomla. Lembre que nós criamos um para abrigar uma imagem para o header.

Este módulo por isso é muito útil.

**Vamos instalar um bom módulo de SlideShow.**

### **Download**

[https://www.joomlack.fr/en/component/dms/view\\_document/59-slideshow-ck](https://www.joomlack.fr/en/component/dms/view_document/59-slideshow-ck)

### **Instalar**

Extensões – Gerenciar – Instalar

Selecionar o módulo

Enviar e Instalar

### **Configurações**

Extensões – Módulos - Sledeshow CK

Oculte o título

Habilite o módulo

Selecione a posição – position-0 (com esta posição precisamos adicionar ao layout)

Atribuir Menu – todas as páginas

Para trocar os Slides – Slides manager

Assim ele aparece em todos os itens de menu. Vamos configurar para que apareça somente na página índice.

Extensões – Módulos - Sledeshow CK

Atribuir Menu – Apenas as páginas selecionadas.

Apenas deixarei marcada o Início

### **Ajustar a Altura**

Styles option

Height – 22% (original é 62%)

Salvar e Fechar

### **Instalar Módulo Popup**

Este módulo pode ser útil algumas vezes para dar algum aviso sobre o site.

Download

<http://www.graphicaholic.com/demos/extensions/popup-aholic>

Instalar

Configurar

- Ocultar título

- Publicar

- Atribuir menu – todas as páginas (se esquecer este, não aparece em nenhuma página)

- Posição – position-1

Salvar e fechar

## **No Layout o Gantry**

Adicionar uma linha abaixo do Navigation para este

## **Instalar o Módulo Pensamento do Dia**

Este pequeno módulo exibe um pensamento selecionado entre 200 outros a cada visita do usuário ou a cada refresh.

Download

[https://ribafs.github.io/joomla/programacao/mod\\_pensamentododia\\_j3.zip](https://ribafs.github.io/joomla/programacao/mod_pensamentododia_j3.zip)

Instalação

Como os anteriores.

Configurações

- Habilitar
- Atribuir menu – Todas as páginas
- Posição – position-1 (não terá importância, basta escolher qualquer um)

Layout no Gantry

Adicionar uma linha abaixo da região Main

Renomear para Pensamentos e vincular ao módulo Pensamento do Dia.

## **Phoca Gallery**

Um dos melhores componentes que conheço é o Phoca Gallery, que cria galerias de imagens.

Download

<http://www.phoca.cz/download/category/1:phoca-gallery-component>

Baixar o componente e o idioma pt-br

1) Instalar ambos

Componentes - Phoca Gallery

Categoria - Novo - Título

Diretório de album - Selecionar diretório - (apenas marcar o diretório na seta verde)

Salvar e Fechar

2) Imagens

Múltipla adição

Selecionar o diretório à direita/abaixo

Carregamento múltiplo

Start Upload (abaixo)

Salvar e Fechar

3) Criar o item de menu

Menus - Hydrogen Theme (ou Main Menu) - Adicionar item de menu

- digitar Título
  - clicar em Selecionar
    - clicar em Phoca Gallery Módulo de Imagem
    - Geralmente a primeira opção
  - Clique na combo Selecionar categoria e selecione a galeria
- Confira no site

## AllVideos

Adição de áudio e vídeo. Outra boa extensão para o Joomla

Resumo do manual

Download

<http://www.joomlaworks.net/extensions/free/allvideos>

Configurar a pasta dos vídeos

/images/audio

/images/video

Crie um novo artigo com as tags de abertura e fechamento

```
{mp4}yourvideofilename(/mp4}
```

Sendo que yourvideofilename.mp4 está na pastas /images/video

Pode usar:

{flv}, {wmv}, {youtube} e {vimeo}

```
{youtube}inTas4CAIs_egj200|150|1{/youtube}
```

For example, an "autoplaying" YouTube video of 100x150 dimensions would be displayed adding this to a content item:

{youtube}inTas4CAIs\_egj200|150|1{/youtube} - First in line is the file name, then the width, then height, and the final "1" indicates autoplay. If you don't want autoplay, make this a "0".

Remoto de outro site:

```
{mp3remote}http://www.filehosting.com/pixies_bonemachine.mp3{/mp3remote}
```

```
{swfremote}http://www.filehosting.com/presentation.swf{/swfremote}
```

```
{mp4remote}http://www.filehosting.com/nadasurf.mp4{/mp4remote}
```



# 15 - Framework CakePHP 3

## 15.1 - Convenções no CakePHP 3

Pode ser um assunto extenso mas compensa a leitura atenta.

A equipe do CakePHP é grande admiradora de convenção sobre configuração. Seguindo convenções você recebe funcionalidades gratuitamente e libera a si mesmo do pesadelo de manter arquivos de configuração.

**Aviso Importante:** Portanto, antes de começar a trabalhar com CakePHP é muito importante conhecer suas convenções para tirar delas as devidas vantagens. Caso não as use o CakePHP não será de muita ajuda.

### Controllers

Nomes de classes tipo Controller devem estar no plural, ser CamelCase e terminarem com o sufixo Controller.

Exemplos de classes controller:

ClientesController, PeopleController e UltimosArtigosController

**Actions** - Métodos public nos controllers são chamados de actions e se comunicam com views com mesmo nome que eles e extensão .ctp.

Um exemplo: index() - index.ctp

o action Controller/ClientesController/index() é mapeado automaticamente para a view src/Template/Clientes/index.ctp.

Métodos protected ou private não podem ser acessados via Routing.

### Considerações sobre URL para nomes de controllers

O ClientesController que está no arquivo ClientesController.php é chamado pelo navegador com:

`http://localhost/aplicacao/clientes`

Nomes de controllers com palavras compostas podem ficar assim:

- /vermelhaMacas
- /VermelhaMacas
- /Vermelha\_macas
- /vermelha\_macas

Todos resolverão com o controller VermelhaMacas.

### **Criar links:**

```
$this->Html->link('link-title', [  
    'prefix' => 'MyPrefix' // CamelCased  
    'plugin' => 'MyPlugin', // CamelCased  
    'controller' => 'ControllerName', // CamelCased  
    'action' => 'actionName' // camelCased  
])
```

### **Namespaces**

Todas as classes do core do CakePHP agora (3.x) usam namespaces e seguem as especificações de autoload (auto-carregamento) do **PSR-4**.

Por exemplo (troca src por Cake e troca as barras)

src/Cache/Cache.php

tem o namespace

Cake\Cache\Cache

Constantes globais e métodos de helpers como `__()` e `debug()` não usam namespaces por questões de conveniência.

### **Nomes de arquivos e Classes:**

Classe `KissesAndHugsController` está no arquivo `KissesAndHugsController.php`

Classe `ClientesController` está no arquivo `ClientesController.php`.

Alguns exemplos de classes e seus arquivos:

- O Controller class `KissesAndHugsController` deve estar no arquivo com nome `KissesAndHugsController.php`
- O Component class `MyHandyComponent` deve estar no arquivo com nome `MyHandyComponent.php`
- A Table class `OptionValuesTable` deve estar no arquivo com nome `OptionValuesTable.php`
- A Entity class `OptionValue` deve estar no arquivo com nome `OptionValue.php`
- O Behavior class `EspecialyFunkableBehavior` deve estar no arquivo com nome `EspecialyFunkableBehavior.php`
- A View class `SuperSimpleView` would deve estar no arquivo com nome `SuperSimpleView.php`
- O Helper class `BestEverHelper` deve estar no arquivo com nome `BestEverHelper.php`

## Model e Bancos de Dados

*Nomes de classes Table* - são no plural e CamelCase

### Chave Primária

Toda tabela, obrigatoriamente deve ter uma chave primária e o nome da chave deve ser id para usufruir das vantagens do CakePHP.

Lembrando que o cake trabalha com nomes em inglês. Ele tem um recurso online importante para mostrar o plural de nomes:

<http://inflector.cakephp.org/>

**Nomes válidos:** Clientes, Populacao, GrandePopulacao e RealmenteGrandePopulacao.

Pesquisando no site acima por Clientes e People, vemos que já estão no plural e seu singular é Cliente e Person

**Nomes de tabelas** são em minúsculas, no plural e palavras compostas separadas por sublinhado. Nomes de tabelas para os acima:

clientes, populacao, grande\_populacao e realmente\_grande\_populacao

A convenção é para usar tabelas e campos com nomes na língua inglesa.

**Nomes de campos** em minúsculas e compostos são separados por sublinhado: first\_name.

Se usarmos os **campos username e password** (com estes nomes) na tabela users, o Cake deve estar apto para auto-configurar algumas coisas para nós, quando implementando o user login.

Obs.: quando usar autenticação use o tamanho do campo password com varchar(255). Também ajuda adicionar um campo chamado role na tabela users.

### Relacionamentos

Chave estrangeira nos relacionamentos hasMany, belongsTo ouhasOne são reconhecidas por default com:

nome da tabela relacionada no singular seguida de "\_id".

Exemplos: groups e users. Em users o campo group\_id para relacionar.

Relacionamento entre articles e users. Em articles adicionar o campo user\_id.

### Relacionamento Muitos para Muitos

Exemplo: Para relacionamento muitos para muitos das tabelas platillos com cocineros, criaremos uma tabela intermediária para relacionar platillos com cocineros:

```
create table cocineros_platillos(  
    id int unsigned auto_increment primary key,  
    cocinero_id int(11) not null,  
    platillo_id int(11) not null  
);
```

## Tipos de Relacionamentos

one to one	hasOne	Um usuario tem um perfil.
one to many	hasMany	Um usuario pode ter múltiplos artigos.
many to one	belongsTo	Muitos artigos pertencem a um usuario
many to many	belongsToMany	Várias Tags pertencem a muitos artigos.

Para uma classe Bakers teremos uma chave estrangeira assim: baker\_id.

Para uma tabela como category\_types a chave estrangeira será category\_type\_id.

Nomes de campos especiais, que levam o Cake a tomar iniciativas importantes para nós:

title

name

created

modified

## Convenções para as Views

As views tem nomes de arquivos em minúsculas com extensão .ctp.

O método getReady() do controller PeopleController está associado ao template/view src/Template/People/get\_ready.ctp.

## Arquivos da Aplicação

Todos os arquivos da aplicação que criamos ficam na pasta src.

## Criptografia

Por padrão o CakePHP 3.x usa a criptografia bcrypt para proteger as senhas.

Documentação oficial

<http://book.cakephp.org/3.0/pt/intro/conventions.html>

## Convenções para as Views

O método getReady() do controller PeopleController está associado ao template/view src/Template/People/get\_ready.ctp.

## Arquivos da Aplicação

Todos os arquivos da aplicação que criamos ficam na pasta src.

## Criptografia

Por padrão o CakePHP 3.x usa a criptografia bcrypt para proteger as senhas.

## 15.2 - Instalação do CakePHP 3

### Pré-requisitos

- Web server como Apache 2 com mod\_rewrite
- PHP 5.6+
- Extensões mbstring e intl
- Um dos 4 SGBDs suportados
- Composer

### Instalação do Composer

- Se tens a versão 7 ou superior do PHP instale assim:  
sudo apt-get install composer

- Para a versão 5.x assim:  
sudo su  
curl -sS https://getcomposer.org/installer | php

Colocar no path do Linux  
mv composer.phar /usr/local/bin/composer

### Instalando o CakePHP/Criar Aplicativo

```
composer create-project --prefer-dist cakephp/app nome_ap
```

Como o comando é grande e pode dar trabalho de memorizar, vamos criar um link (no Windows crie um bat com %1), no Linux para receber parâmetro é \$1.  
sudo nano /usr/local/bin/comp

Cole a linha  
composer create-project --prefer-dist cakephp/app \$1

```
sudo chmod +x /usr/local/bin/comp
```

### Observação importante

Criei dois scripts sql para que tudo ande como planejado. Use estes scripts caso tenha insegurança do que está fazendo.

### Execute assim:

```
cd /var/www/html
```

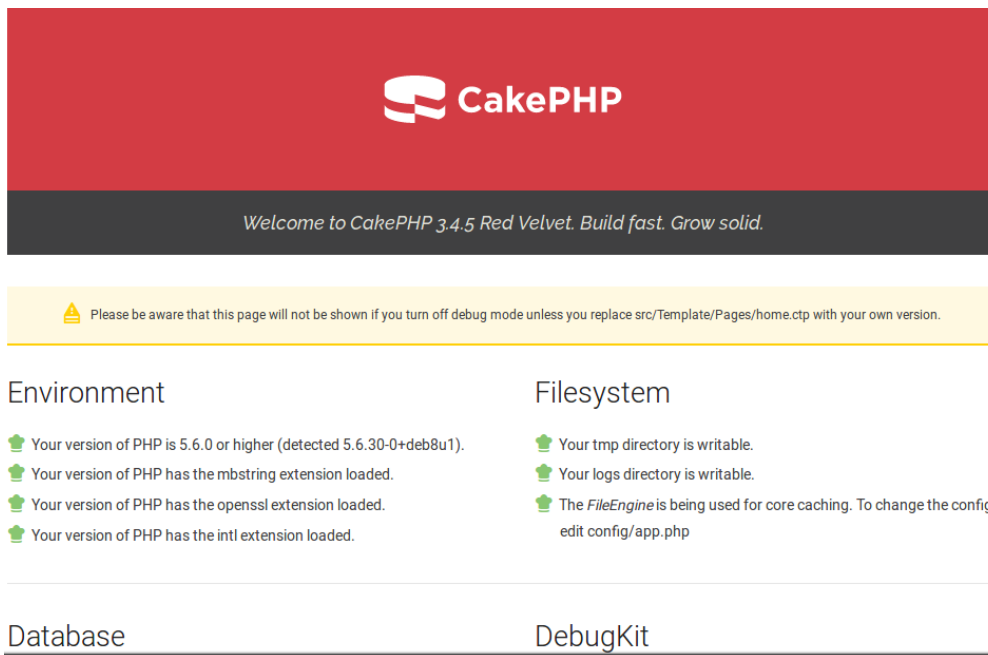
```
comp cliente
```

Ao finalizar o Cake já terá trazido todo o seu código para a pasta /var/www/html/cliente. E já vem com um controller default para que vejamos algo.

## Chame pelo navegador

<http://localhost/cliente>

Aparecerá então a tela abaixo:



## Configurações do banco e do routes em config/app.php

Na primeira vez que seguir este roteiro, caso tenha insegurança, crie o banco de dados e importe o script que estou apresentando.

Para criar um aplicativo de testes, apenas para testar este roteiro, estou oferecendo dois script de bancos de dados (clientes), um para mysql e outro para postgresql.

Crie o banco chamado **clientes** usando um destes scripts.

### Configuração do Banco

Abra o script config/app.php e faça as alterações abaixo.

Se for usar o MySQL basta alterar estas linhas:

```
'username' => 'root',  
'password' => 'mysql',  
'database' => 'estoque',
```

Se for usar o PostgreSQL, precisa mudar mais esta linha, que fica acima das 3:

```
'driver' => 'Cake\Database\Driver\Postgres',
```

E adicionar abaixo das 3, mas somente se usar esquemas:

```
'schema' => 'clientes',
```

## **Routes**

Para que ao abrir o aplicativo automaticamente ele mostra o controller Clientes

Em config/routes.php altere a linha abaixo:

Originalmente é assim a linha:

```
$routes->connect('/', ['controller' => 'Pages', 'action' => 'display', 'home']);
```

Mudei para

```
$routes->connect('/', ['controller' => 'Clientes', 'action' => 'index']);
```

## **Geração de Código com o Bake**

Depois de configurados o app.php e o routes.php. O script criou 3 tabelas.

```
cd cliente
```

```
bin/cake bake all groups
```

```
bin/cake bake all users
```

```
bin/cake bake all clientes
```

## **Sabendo a versão do CakePHP instalada pelo seu console**

```
bin/cake
```

Mostrará algumas informações e esta linha:

```
Welcome to CakePHP v3.4.5 Console
```

## Acesse pelo Navegador

Agora pode acessar seu aplicativo pelo navegador

<http://localhost/clientes>

Aparece por default o Clientes/index, como configuramos no routes

Id	Name	Birthday	Phone	User	Created	Modified	Actions
10	Rudyard Weber	10/26/15	(445) 457-4552	1			View Edit Delete
13	Dustin M. Oneil	4/24/16	(276) 722-0976	1			View Edit Delete
19	Wesley Garner	6/11/16	(578) 231-2389	1			View Edit Delete
20	Irene P. Arnold	2/12/16	(253) 631-9830	1			View Edit Delete
22	Roanna K. Ortiz	8/28/16	(336) 788-2145	1			View Edit Delete
29	Flynn B. Willis	10/30/16	(898) 157-2508	1			View Edit Delete

Id	Name	Created	Modified	Actions
1	Supers	8/30/16, 9:15 PM	8/30/16, 9:15 PM	View Edit Delete
2	Admins	8/30/16, 9:15 PM	8/30/16, 9:15 PM	View Edit Delete
3	Managers	8/30/16, 9:15 PM	8/30/16, 9:15 PM	View Edit Delete
4	Users	8/30/16, 9:15 PM	8/30/16, 9:15 PM	View Edit Delete

< previous next >

Page 1 of 1, showing 4 record(s) out

Id	Username	Password	Group	Created	Modified	Actions
1	super	\$2y\$10\$DPJ0QvPAOpae23WWWVmVD5PpRJIJdCa/BUJdu4eZWoX6Hu98aTdnq	Supers	9/15/16, 3:57 PM	9/15/16, 3:57 PM	View Edit Delete
2	admin	\$2y\$10\$2aTDwG Etcayt859IMTuWN.Bys.5Mpmii5pIN2oOkfzYbgGIM7v9q6	Admins	9/15/16, 3:57 PM	9/15/16, 3:57 PM	View Edit Delete
3	manager	\$2y\$10\$08jY05pe/fb9rd.JelizQZexgb6czFARx7ZZZ8tp4rcGzhvgrloPLY	Managers	9/15/16, 3:57 PM	9/15/16, 3:57 PM	View Edit Delete

Agora está pronto para melhorar o aplicativo manualmente ou usando um plugin como o cake-control-br.



## Alterando o Template/View

Quando acessar pelo navegador, especificamente users, aparece o campo senha com o hash. Não é interessante que este campo apareça, então vamos removê-lo da view.

Editar então o arquivo:

```
cd /var/www/html/clientes
cd src/Template/Users/index.ctp
```

Apagar as duas linhas:

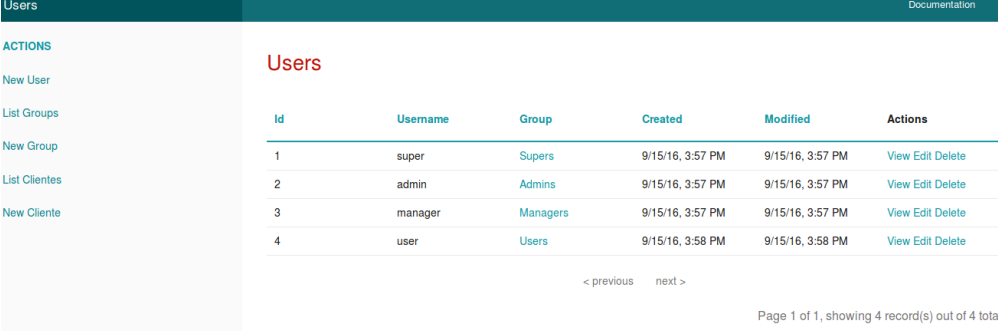
```
<th scope="col"><?= $this->Paginator->sort('password') ?></th>
```

e

```
<td><?= h($user->password) ?></td>
```

Salve e visualize novamente.

Agora está melhor, veja.



Id	Username	Group	Created	Modified	Actions
1	super	Supers	9/15/16, 3:57 PM	9/15/16, 3:57 PM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
2	admin	Admins	9/15/16, 3:57 PM	9/15/16, 3:57 PM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
3	manager	Managers	9/15/16, 3:57 PM	9/15/16, 3:57 PM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
4	user	Users	9/15/16, 3:58 PM	9/15/16, 3:58 PM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

< previous next >

Page 1 of 1, showing 4 record(s) out of 4 total

## 15.3 - Usando o Plugin cake-control-br do CakePHP 3

### Criar o Aplicativo

Criaremos um novo aplicativo, com um novo banco, já que o cake-control-br requer umas tabelas básicas (groups, users e permissions).

```
cd /var/www/html  
comp cliente2
```

Com isso ele gera aquela página default que vimos no tutorial de Instalação

### Criação do Banco de Dados

Criamos um novo aplicativo e precisamos de outro banco de dados. Criaremos o banco cliente2 usando o script do plugin cake-control-br. Fazemos isso agora, antes de continuar

### Instalando o plugin cake-control-br

```
cd cliente2  
composer require ribafs/cake-control-br
```

### Habilite o plugin com:

```
bin/cake plugin load CakeControlBr --bootstrap
```

### Download do plugin

<https://github.com/ribafs/cake-control-br>

### Configurações

Configure o banco de dados em config/app.php (ajuste as linhas do banco) e o roteamento em config/routes.php (Troque por Customers/index)  
Veja o tutorial sobre Instalação se tiver alguma dúvida

### Descompactar e abrir a pasta docs para copiar sobrescrevendo:

- bootstrap\_cli.php para a clientes/config (Bake gerará Users com login e logout)
- AppController.php para clientes/src/Controller (diversas alterações)

## Gerar o Código com o Bake

```
cd cliente2
```

```
bin/cake bake all groups -t CakeControlBr  
bin/cake bake all users -t CakeControlBr  
bin/cake bake all permissions -t CakeControlBr  
bin/cake bake all customers -t CakeControlBr
```

Assim, temos 4 usuários:

super – todos os poderes

admin – somente tabelas administrativas

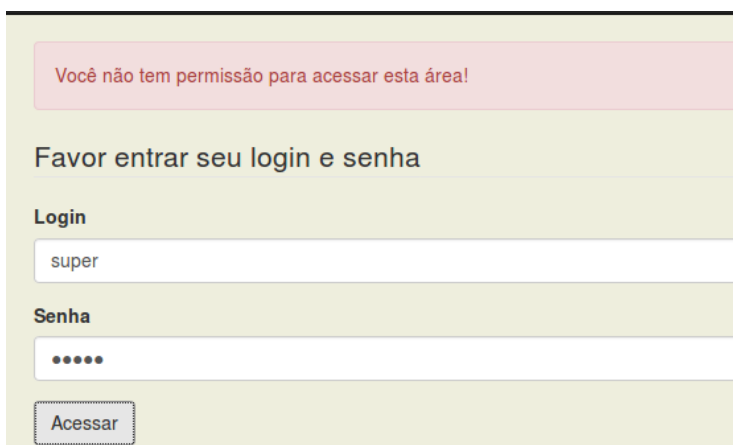
manager – somente tabelas do front

user – nenhum poder, apenas o de logar.

## Veja como fica nosso aplicativo

Agora veremos telas customizadas e com o framework Bootstrap, tudo traduzido, com controle de acesso e outros bons recursos.

### Login



Você não tem permissão para acessar esta área!

Favor entrar seu login e senha

Login

Senha

Acessar

## Permissões

## Customers

Id	Name	Birthday	Phone	User	Created	Modified	Ações
10	Rudyard Weber	26/10/2015	(445) 457-4552	1			
13	Dustin M. O'Neill	24/04/2016	(276) 722-0976	1			
19	Wesley Garner	11/06/2016	(578) 231-2389	1			
20	Irene P. Arnold	12/02/2016	(253) 631-9830	1			
22	Roanna K. Ortiz	28/08/2016	(336) 788-2145	1			
29	Flynn B. Willis	30/10/2016	(898) 157-2500	1			
34	Jessamine Bright	10/10/2015	(659) 268-8156	1			
38	Amos Holcomb	13/06/2016	(320) 249-8062	1			
42	Rashad S. Le	13/04/2016	(469) 302-1629	1			
46	Jorden Galloway	11/09/2016	(930) 855-8859	1			

## Users

Id	Username	Password	Group	Created	Modified	Ações
1	super	\$2y\$10\$DPJQQvPAOpa23WWWmVQ.5PsRJIUdCa/BUdu4eZW0X6Hu98aTdnq	Supers	15/09/2016 15:57:16	15/09/2016 15:57:16	
2	admin	\$2y\$10\$2aTDwGEtcayl859IMTuwn.Bys.5Mpmi5plN2oQkFzYbgGIM7v9q6	Admins	15/09/2016 15:57:28	15/09/2016 15:57:28	
3	manager	\$2y\$10\$08jYO5peib9rd.JetizQZexgb6czFARx72ZZ8tp4rcGzvhgrloPLY	Managers	15/09/2016 15:57:39	15/09/2016 15:57:39	
4	user	\$2y\$10\$pxNdOGydRKprWn/levNz/eyoUxa/zXONDj29RelIXdCqu.LlxjYb6	Users	15/09/2016 15:58:21	15/09/2016 15:58:21	

O mesmo problema que no anterior aplicativo, mostrando o hash das senhas. Remove na view respectiva como fizemos no tutorial de Instalação.

Ficará assim:

Id	Username	Group	Created	Modified	Ações
1	super	Supers	15/09/2016 15:57:16	15/09/2016 15:57:16	
2	admin	Admins	15/09/2016 15:57:28	15/09/2016 15:57:28	
3	manager	Managers	15/09/2016 15:57:39	15/09/2016 15:57:39	
4	user	Users	15/09/2016 15:58:21	15/09/2016 15:58:21	

## 15.4 - Arquivos em Geral do CakePHP 3

Vamos dar uma olhada na estrutura de diretórios do CakePHP 3.

### Estrutura de Arquivos e Diretórios

cliente/bin – script executável cake for linux e for windows,

cliente/config – os vários scripts de configuração, app.php, routes.php, etc

cliente/logs

cliente/plugins – plugins de terceiros instalados manualmente

cliente/src – todo o código da aplicação fica neste diretório: controllers, models, views, etc.

cliente/tests

cliente/tmp

cliente/vendor – todas as bibliotecas, do CakePHP e de terceiros

cliente/webroot – css, javascript, imagens, etc

cliente/composer.json

cliente/composer.lock

cliente/index.php

cliente/phpunit.xml.dist

cliente/README.md

Mais detalhes:

<https://book.cakephp.org/3.0/pt/installation.html>

## 15.5 - Gerando Código com Bake no CakePHP 3

<http://book.cakephp.org/3.0/en/bake.html>

A console bake do CakePHP é um outro esforço para que você esteja rodando o CakePHP rapidamente. A console do bake pode criar qualquer um dos códigos básicos do CakePHP: models, behaviors, views, helpers, controllers, components, test cases, fixtures e plugins. E não estamos falando apenas de esqueletos de classes, o bake pode criar uma aplicação inteiramente funcional em poucos minutos.

### Pré-requisitos

A console do bake requer:

- PHP CLI
- extensão intl
- Pelo menos um SGBD instalado e configurado na aplicação atual

### Instalação

O bake já vem instalado por padrão, mas se precisar instalar use o comando:  
`composer require --dev cakephp/bake:~1.0`

ou

```
php composer.phar require --dev cakephp/bake:dev-master
```

Comandos disponíveis do bake. Execute:

```
bin\cake bake
```

- all
- behavior
- cell
- component
- controller
- fixture
- form
- helper
- model
- plugin
- shell
- template
- test

Usando ``cake bake [name]`` podemos invocar uma tarefa especificada pelo name.

Exemplo:

```
bin\cake bake all [name]
```

Ajuda

bin/cake bake controller --help

O bake é uma ferramenta espetacular. Veja que com apenas um comando geramos o aplicativo. Com ele podemos fazer muito:

Gerar um controller:

bin/cake bake controller Clientes

Gerar controller sem actions:

bin\cake bake controller --no-actions Clientes

Gerar Componente

bin/cake bake component Calculo

Gerar um Template

bin/cake bake template Clientes

Gerar Helper

bin/cake bake helper MeuForm

Gerar Cell

bin/cake bake cell MinhaCelula

Gerar Form

bin/cake bake form MeuForm

Gerar um Model

bin/cake bake model Clientes

Gerar Behavior

bin/cake bake behavior MeuBehavior

Observe as mensagens para saber em que diretório está sendo criado.

Ajuda

bin/cake bake controller --help

Existem alguns recursos que não são gerados ou auxiliados pelo bake, como o element.

**Geração de todo o código: controllers, models e templates e testes**

bin/cake bake all clientes

bin/cake bake all users

**Apenas controller, model e template**

bin/cake bake controller clientes

bin/cake bake model clientes

bin/cake bake template clientes

## Plugins

Criar plugin ContactManager com tabela contacts

```
cd /backup/www/cake/blog
bin/cake bake plugin ContactManager
bin/cake bake controller --plugin ContactManager Contacts
bin/cake bake model --plugin ContactManager Contacts
bin/cake bake template --plugin ContactManager Contacts
```

## Atualizando

```
composer dumpautoload
```

## Componente

```
bin\cake bake component Calculo
```

## Migrate

```
bin/cake migrations
bin/cake migrations migrate
bin/cake migrations status --plugin PluginName
```

## Criar plugin usando o bake:

O banco precisa ter uma tabela chamada Contacts

```
cd /backup/www/cake/blog
bin/cake bake plugin ContactManager
bin/cake bake controller --plugin ContactManager Contacts
bin/cake bake model --plugin ContactManager Contacts
bin/cake bake template --plugin ContactManager Contacts
```

## Gerando código para Área Administrativa

Criar aplicativo para cadastro de clientes com área administrativa acessada somente por alguns usuários.

## Criar o controller para o Admin:

```
bin\cake bake controller --no-actions --prefix Admin Home
```

## Criar o controller para o Site

```
bin\cake bake controller --no-actions --prefix Site Home
```



## **Criar um CRUD para o Admin:**

Configure o banco de dados no config/app.php

Criar código para gerar classe

```
bin\cake bake migration CreateUsers nome:string email:string created modified
```

## **Criar tabela com código acima**

```
bin\cake migrations migrate
```

## **Gerar o Scaffold para Usuários**

```
bin\cake bake all Users --prefix Admin
```

## **Criar outros usos para o Bake:**

Configurar o banco de dados

Configurar a carga de plugin

Criar:

- model
- controller

```
bin/cake bake controller comments -force  
ou
```

```
bin/cake bake controller comments -f
```

- controller sem actions
- action
- view/template
- helper
- element
- component
- criar validações interativamente
- mudar/criar displayField
- Criar admin

```
bin/cake bake template comments -f
```

## **Usar migrações**

Criar migrações

```
bin\cake migrations create CreateUsersTable
```

```
bin\cake migrations create CreateGroupsTable
```

Criar tabelas

```
bin\cake migrations migrate
```

## **Usar Composer**

## Instalar CakePHP

```
composer create-project --prefer-dist cakephp/app nopeapp
```

```
chmod 550 bin\cake
```

## Garantir que não use cache:

Você deve precisar regenerar suas classes model e limpar o cache:

```
bin/cake bake model comentarios --force
```

```
bin/cake bake model erros --force
```

```
bin/cake orm_cache clear
```

## Customizar o Bake

Se quisermos customizar o emplate do bake apenas para um certo controler, como o

Servidores, para que o bake gere para o mesmo somente os actions: index(), view() e add(), alteramos o código do config/bootstrap\_cli.php para algo assim:

```
<?php
use Cake\Event\Event;
use Cake\Event\EventManager;
use Cake\Utility\Hash;

EventManager::instance()->attach(function (Event $event) {
    $view = $event->subject;
    $name = Hash::get($view->viewVars, 'name');
    $isController = strpos($event->data[0], 'Bake/Controller/controller.ctp') !== false;
    if ($isController !== false && $name == 'Servidores') {
        $view->viewVars['actions'] = ['index', 'view', 'add'];
    }
    if ($isController && $name == 'Comentarios') {
        $view->viewVars['actions'] = ['add'];
    }
}, 'Bake.beforeRender'); ?>
```

Este código deve:

- Anexar ao evento Bake.beforeRender, que permite modificar qualquer dado indo

para o template/view

- Receber o nome do template
- Caso o nome do arquivo termine com Bake/Controller/controller.ctp, o template usado sendo o Servidores, gerará apenas os referidos actions.
- Também adicionamos somente o action add() caso o controller seja Comentários.

Mais detalhes no artigo do José Gonzales em:

<http://josediazgonzalez.com/2014/12/03/customizing-bake-and-installing-plugins-with-cakephp-3/>

## 15.6 - Dicas

### Data e Hora com CakePHP 3

```
class Cake\I18n\Time
```

Se você precisa das funcionalidades do TimeHelper fora de uma View, então use a classe Time:

```
use Cake\I18n\Time;
```

```
class UsersController extends AppController
{
    public function initialize()
    {
        parent::initialize();
        $this->loadComponent('Auth');
    }

    public function afterLogin()
    {
        $time = new Time($this->Auth->user('date_of_birth'));
        if ($time->isToday()) {
            // Greet user with a happy birthday message
            $this->Flash->success(__('Happy birthday to you...'));
        }
    }
}
```

CakePHP usa Chronos para dar poder ao seu utilitário Timer. Qualquer coisa que você possa fazer com Chronos e DateTime você pode fazer com Time e Date:

<https://github.com/cakephp/chronos>

<http://api.cakephp.org/chronos/1.0/>

Antes da versão 3.2 o CakePHP usava Carbon:

<https://github.com/briannesbitt/Carbon>

### Criando Instâncias de Time

Existem algumas maneiras de criar instâncias de Time:

```
use Cake\I18n\Time;
```

```
// Create from a string datetime.
$time = Time::createFromFormat(
    'Y-m-d H:i:s',
    $datetime,
    'America/Fortaleza'
);
```

```
// Create from a timestamp
$time = Time::createFromTimestamp($ts);
```

```
// Get the current time.
```

```

$time = Time::now();

// Or just use 'new'
$time = new Time('2014-01-10 11:11', 'America/New_York');

$time = new Time('2 hours ago');

Mais:
// Fixate time.
$now = new Time('2014-04-12 12:22:30');
Time::setTestNow($now);

// Returns '2014-04-12 12:22:30'
$now = Time::now();

// Returns '2014-04-12 12:22:30'
$now = Time::parse('now');

```

## Manipulação

Depois de criada a instância podemos manipular as instâncias de Time usando métodos setters.

```

$now = Time::now();
$now->year(2016)
    ->month(08)
    ->day(31);

```

Também podemos usar os métodos providos pelo PHP embutidos na classe DateTime:

```

$now->setDate(2016, 08, 31);

```

Datas podem ser modificadas através da subtração e adição de seus componentes:

```

$now = Time::now();
$now->subDays(5);
$now->addMonth(1);

// Using strtotime strings.
$now->modify('+5 days');

```

Você pode receber os componentes internos da data acessando suas propriedades:

```

$now = Time::now();
echo $now->year; // 2014
echo $now->month; // 5
echo $now->day; // 10
echo $now->timezone; // America/Fortaleza

```

Também é permitido assinalar diretamente essas propriedades para modificar a data:

```

$time->year = 2015;
$time->timezone = 'Europe/Paris';

```

## Formatando

```

static Cake\I18n\Time::setJsonEncodeFormat($format)

```

O método configura o formato default usado quando convertendo um objeto para json:  
`Time::setJsonEncodeFormat('yyyy-MM-dd HH:mm:ss');`  
Este método precisa ser chamado estaticamente.

```
Cake\I18n\Time::i18nFormat($format = null, $timezone = null, $locale = null)
```

Algo bem comum a fazer com instâncias de Time é imprimir as datas formatadas. Veja:

```
$now = Time::parse('2014-10-31');  
  
// Prints a localized datetime stamp.  
echo $now;  
  
// Outputs '10/31/14, 12:00 AM' for the en-US locale  
$now->i18nFormat();  
  
// Use the full date and time format  
$now->i18nFormat(\IntlDateFormatter::FULL);  
  
// Use full date but short time format  
$now->i18nFormat([\IntlDateFormatter::FULL, \IntlDateFormatter::SHORT]);  
  
// Outputs '2014-10-31 00:00:00'  
$now->i18nFormat('yyyy-MM-dd HH:mm:ss');
```

### Configurando o Locale Padrão e o Formato de String

```
// The same method exists on Date, FrozenDate, and FrozenTime  
Time::setDefaultLocale('pt-BR');
```

### Datas

A classe Date no CakePHP implementa a mesma API e métodos que o Time\I18n\ . A principal diferença entre hora e data é que Data não acompanha componentes de tempo, e está sempre em UTC. Como um exemplo:

```
use Cake\I18n\Date;  
$date = new Date('2015-06-15');  
  
$date->modify('+2 hours');  
// Outputs 2015-06-15 00:00:00  
echo $date->format('Y-m-d H:i:s');  
  
$date->modify('+36 hours');  
// Outputs 2015-06-15 00:00:00  
echo $date->format('Y-m-d H:i:s');
```

Mais em:

<http://book.cakephp.org/3.0/en/views/helpers/time.html>

<http://book.cakephp.org/3.0/en/core-libraries/time.html>

<http://book.cakephp.org/3.0/en/core-libraries/internationalization-and-localization.html#parsing-localized-dates>

## Máscaras no CakePHP

- Máscara na view index (CPF)

Enquanto não descubro uma função do Cake que faz isso...

O CPF é um campo texto mas constituído somente de números. A listagem (index.ctp) mostra a relação de números sem nenhuma formatação. Ajudaria se formatássemos usando uma máscara adequada para CPF, ajudaria a visualizar. Vamos fazer isso agora para a view Clientes/index.ctp:

Substitua a linha

```
<td><?php echo h($cliente['Cliente']['cpf']); ?>&nbsp;  </td>
```

Por este código:

```
<?php
$cpfmask1 = substr($cliente['Cliente']['cpf'], 0,3);
$cpfmask2 = substr($cliente['Cliente']['cpf'], 3,3);
$cpfmask3 = substr($cliente['Cliente']['cpf'], 6,3);
$cpfmask4 = substr($cliente['Cliente']['cpf'], 9,2);
$cpfmask = $cpfmask1.'.'. $cpfmask2.'.'. $cpfmask3.'-'. $cpfmask4
?>
```

```
<td><?php echo h($cpfmask); ?>&nbsp;  </td>
```

Com isso a listagem mostrará o CPF com a sua máscara.

- Máscara em formulários com jQuery

Adicionando Máscaras com o plugin maskedinput do jQuery:

- Baixar o jQuery - <http://jquery.com/>

Maskedinput - <http://digitalbush.com/projects/masked-input-plugin/>

- Renomear jQuery para jquery.js e maskedinput para jquery.maskedinput.js

- Copiar ambos para app/webroot/js

- Criar a referência para ambos em app/View/Layout/default.ctp ou outro layout, na seção head, assim:

```
echo $this->Html->script(array('jquery','jquery.maskedinput'));
```

- Editar a view onde deseja a máscara e adicionar ao final:

```
<script type="text/javascript">
    jQuery(document).ready(function($){
        $("#cpf").focus();
        $("#cpf").mask("999.999.999-99");
    });
</script>
```

Ainda na mesma view edite o campo onde ficará a máscara:

```
echo $this->Form->input('cpf',array('label'=>'CPF','id'=>'cpf'));
```

Créditos: <http://linguagensdeprogramacao.wordpress.com/2011/11/16/mascaras-com-cakephp/>

Quando customizamos um campo para exibir máscara, precisamos remover a máscara antes de mandar para o banco, especialmente campos numéricos.

## 15.7 - Projeto com GitHub e Packagist

Ainda tem um terceiro projeto envolvido, o Composer.  
O Composer funciona de forma transparente entre ambos

### Criando um arquivo `composer.json`

Projeto em PHP na dobradinha GitHub e Packagist

Ainda tem um terceiro projeto envolvido, o Composer.  
O Composer funciona de forma transparente entre ambos

### GitHub

- Criar conta
- Criar novo repositório
- Enviar código fonte para o repositório

### 1) Crie seu projeto em PHP

- Use um código existente
- Crie um novo
- Faça um fork de algum projeto do GitHub e customise

### 2) Criando a estrutura do seu projeto

```
2.1) cd /var/www/html
    mkdir meu-projeto
    cd meu-projeto
    composer init
```

### Responder as perguntas

Package name (<vendor>/<name>) [ribafs/meu-projeto]: (apenas tecle enter para aceitar)

Description []: Meu Primeiro projeto com o Packageist

Author [, n to skip]: Ribamar FS <ribafs@gmail.com>

Minimum Stability []: dev

Opções: dev, alpha, beta, RC ou stable

Package Type (e.g. library, project, metapackage, composer-plugin) []: project

License []: MIT

Define your dependencies.

Would you like to define your dependencies (require) interactively [yes]? no

(Nesta primeira etapa - no)

Would you like to define your dev dependencies (require-dev) interactively [yes]? no

```
{
  "name": "ribafs/meu-projeto",
```



```
"description": "Meu Primeiro projeto com o Packageist",
"type": "project",
"license": "MIT",
"authors": [
  {
    "name": "Ribamar FS",
    "email": "ribafs@gmail.com"
  }
],
"minimum-stability": "dev",
"require": {}
}
```

Do you confirm generation [yes]?

### **Concluído**

Referência sobre o composer - <https://getcomposer.org/doc/04-schema.md>

A estrutura do projeto deve ficar assim (sugestão/recomendação):

```
/meu-projeto
  /src (código fonte do meu projeto)
  /vendor (diretório criado pelo Composer para armazenar as dependências)
  composer.json
  README.md
  index.php
```

### **3) Abrigar meu-projeto no GitHub**

- Criar uma conta no GitHub - <https://github.com/>
- Criar um repositório para meu-projeto

### **Adicionar uma chave SSH**

**Em seu desktop Linux execute:**

```
cd
ssh-keygen -t rsa -b 4096 -C "contato@ribafs.org"
Enter
Enter
Enter
```

Checando a senha

```
ls -al ~/.ssh
```

### **Capturando o hash da senha**

```
cat ~/.ssh/id_rsa.pub
```

Copie a chave mostrada desde ssh-rsa ... até seu e-mail, incluso.

Vá até o GitHub, faça login e clique acima em sua foto e Settings  
SSH and GPG keys  
New SSH key

Entre com um Title (Residência)

E cole na caixa Key abaixo

E clique em Add SSH key

Ao criar um repositório no GitHub ele recebe o estágio de desenvolvimento, com um nome:  
master-dev

Nesta fase informamos no composer.json assim:

```
"minimum-stability": "dev",
```

Quando considerarmos que já está usável e queremos oferecer para o mundo, então  
distribuimos assim:

```
composer require ribafs/cake-control:dev-master
```

Ou

editando o composer.json do raiz do aplicativo para adicionar a linha abaixo na seção require:

```
"ribafs/cake-control": "master-dev"
```

No Packagist ele aparece como  
dev-master

Quando considerar que o projeto está estável crie um Release e dê um número para o mesmo.  
Agora pode oferecer duas versões: a estável e a em desenvolvimento, assim:

Dev

```
composer require ribafs/cake-control:dev-master
```

Estável

```
composer require ribafs/cake-control-twbs
```

Depois que configuramos o Packagist e sincronizamos as atualizações somente precisamos interagir  
com

nosso código no GitHub, o Packagist e o Composer ficam transparentes.

#### 4) Enviar meu-projeto para o repositório do GitHub

```
cd /var/www/html/meu-projeto
```

```
git config --global user.email "contato@ribafs.org"  
git config --global user.name "Ribamar FS"  
git init  
git add .  
git commit -m "Primeiro commit"  
git remote add origin git@github.com:ribafs/cake-control.git  
git push -f origin master
```

Para enviar novamente após alterações  
remover a pasta .git  
e repetir os passos acima

O projeto está agora hospedado no GitHub e pronto para ser abrigado pelo Packagist.

#### 5) Criando o Pacote no Packagist

Acesse <https://packagist.org/>

Crie uma conta

Faça login

Clique em Submit a Package

Cole a linha a seguir, lembrando que o nome do pacote deve ser em minúsculas, palavras compostas separadas por hífen:

```
git@github.com:ribafs/cake-control.git
```

E clique em Check

Se retornar correto aceite e faça o submit

#### 6) Atualizações

Cada vez que atualizar o projeto no GitHub precisa atualizar no Packagist

- Vá para sua conta no packagist
- Settings
- My Packages
- Clique no projeto
- Clique em Update (botão verde)

##### **Para tornar automáticas as atualizações**

- Vá para seu repositório no GitHub
- Clique em Settings
- Integration & services
  - Services - Add service - packagist
  - User - ribafs

- Token - o token da sua API (Acesse o site do packagist - login - Profile e copie o token gerado e cole aqui)

- Domain - packagist.org

- Marque Active

Clique em Update service abaixo

Clique acima e à direita em Test service

Caso receba a mensagem "Okay, the test payload is on its way." já poderá ir ao Packagist e abrir seu projeto que não

mais aparecerá a mensagem de que precisa atualizar.

Agora as atualizações foram automatizadas.

Você está pronto para compartilhar seu projeto com o mundo.

Para instalar um plugin (ex.: cake-control), crie um aplicativo em Cake com composer

cd aplicativo

composer require ribafs/cake-control

Mas, desta forma somente será possível após criar um release do projeto no GitHub.

Enquanto não conclue satisfatoriamente seu projeto, poderá divulgar para testes assim:

Editar o composer.json:

Adicionar na seção require-dev a linha

```
"require-dev": {  
    "psy/psysh": "@stable",  
    "cakephp/debug_kit": "~3.2",  
    "cakephp/bake": "~1.1",  
    "ribafs/cake-control": "master-dev"  
},
```

Depois executar:

```
composer update
```

Ou então diretamente para linha de comando:

```
composer require ribafs/cake-control:dev-master
```

### **Como remover plugin:**

Removível somente se bem inicialmente.

```
composer remove cakephp/localized
```

Remover manualmente a linha no bootstrap.php e todo o código adicionado ao vendor além da linha respectiva no composer.json

## Referências:

Composer - <https://getcomposer.org/doc/03-cli.md>

<http://blog.jgrossi.com/2013/creating-your-first-composer-packagist-package/>

Markdown tutorial - <http://www.markdowntutorial.com/> (a linguagem usada pelo GitHub)

## Dicas úteis

<https://gist.github.com/>

## Criação de plugins para o CakePHP3

```
comp app_control
```

Criar um plugin:

```
cd app_control
```

```
bin/cake bake plugin AdminApp
```

Carregar o plugin no bootstrap.php:

```
bin/cake plugin load AdminApp
```

Estrutura de um plugin:

```
/src
```

```
/plugins
```

```
  /ContactManager
```

```
    /config
```

```
    /src
```

```
      /Controller
```

```
        /Component
```

```
      /Model
```

```
        /Table
```

```
        /Entity
```

```
        /Behavior
```

```
      /View
```

/Helper  
/Template  
/Layout  
/tests  
/TestCase  
/Fixture  
/webroot

Criando um controller para o plugin:

```
bin/cake bake controller --plugin AdminApp Groups
```

```
bin/cake bake controller --plugin AdminApp Users
```

```
bin/cake bake controller --plugin AdminApp Permissions
```

**Criando um componente:**

```
bin/cake bake component --plugin AdminApp Control
```

Criar o banco e configurar no app.php e o routes para Groups/index

**Criando um model para o plugin:**

```
bin/cake bake model --plugin AdminApp Groups
```

```
bin/cake bake model --plugin AdminApp Users
```

```
bin/cake bake model --plugin AdminApp Permissions
```

Criando templates para o plugin:

```
bin/cake bake template --plugin AdminApp Groups
```

```
bin/cake bake template --plugin AdminApp Users
```

```
bin/cake bake template --plugin AdminApp Permissions
```

Criando a view login.ctp para o plugin:

```
bin/cake bake template --plugin AdminApp Users login
```

Chamando o plugin pelo navegador:

`http://localhost/app_control/admin-app/groups`

Dica: Sempre que criar um plugin regere o autoloader:

`php composer.phar dumpautoload`

O routes é criado automaticamente.

Criar um helper para o plugin:

`bin/cake bake helper --plugin MeuHelper`

Criando um Behaviour para o plugin:

`bin/cake bake behavior --plugin MeuBehavior`

Routes

```
Router::scope('/adminapp', ['plugin' => 'AdminApp'], function ($routes) {  
    $routes->fallbacks();  
});
```

## 15.8 - Implementação de busca no CakePHP 3

Adaptação de:

<http://www.tayron.com.br/blog/121/criando-um-formulario-de-pesquisa-com-cakephp3>

O exemplo aqui apresentado foi com um aplicativo criado através do bake.

Apenas duas tabelas:

```
CREATE TABLE IF NOT EXISTS `groups` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(100) NOT NULL,  
  `created` datetime DEFAULT NULL,  
  `modified` datetime DEFAULT NULL,  
  PRIMARY KEY (`id`)  
);
```

```
CREATE TABLE IF NOT EXISTS `users` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `username` varchar(255) NOT NULL,  
  `password` char(255) NOT NULL,  
  `group_id` int(11) NOT NULL,  
  `created` datetime DEFAULT NULL,  
  `modified` datetime DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `username` (`username`)  
);
```

Banco - busca

Diretório - busca

Os dados do formulário deve ser enviado via query string (get).

E a action do formulário deve sempre apontas para o método index, exemplo:

`www.site.com.br/users/index.`



View Users/index.ctp

Entendido as duas regras acima vamos criar nosso formulário na nossa view antes da tag <table>:

```
<?php
    echo $this->Form->create(null, ['type' => 'get']);
    echo $this->Form->input('search',
        ['class' => 'form-control', 'label' => false,
        'placeholder' => 'Digite o username',
        'value' => $this->request->query('search')]);
    echo $this->Form->button('Pesquisar');
    echo $this->Form->end();
?>
<hr />
```

Substitua o método index() existente do Controller/UserController.php por:

```
public function index()
{
    $this->paginate = [
        'contain' => ['Groups'],
        'conditions' => ['and' => [
            //Users.people like' => '%' . $this->request->query('search') . '%',
            'Users.username like' => '%' . $this->request->query('search') . '%'
        ]],
        'order' => ['Users.id' => 'DESC' ]
    ];

    $this->set('users', $this->paginate($this->Users));
    $this->set('_serialize', ['users']);
}
```

Implementada no Plugin <https://github.com/ribafs/cake-control-br>

## 15.9 – MVC no CakePHP 3

### Cake na Prática

#### Controller e Views

No controller src/Controller/CientesController.php

```
public function index()
{
    $this->paginate = [
        'contain' => ['Users']
    ];
    $this->set('clientes', $this->paginate($this->Clientes));
    $this->set('_serialize', ['clientes']);

    // Testes
    $this->set('cor', 'Verde');
    $data = [
        'color' => 'pink',
        'type' => 'sugar',
        'base_price' => 23.95
    ];

    // Make $color, $type, and $base_price available to the view:
    $this->set($data);

    $this->set($data);
    $cor1='verde';
    $cor2='azul';
    $this->set(compact('cor1','cor2'));
}
```

Na view src/Template/Cientes/index.ctp

```
<?php
echo 'Cor: '.$cor.<br>';
echo $color.<br>';
echo $type.<br>';
echo $base_price.<br><br>';

echo $cor1.<br>';
echo $cor2.<br><br>';

?>
```

Redirect

Em um action:

```
return $this->redirect(['controller'=>'Clientes', 'action'=>'add']);
```

Na versão 3 do Cake, os Table Objects contém:

queries  
callbacks  
validations  
events

### Passando variáveis pela URL para uma view

Na URL:

localhost/cake-control-demo/pages/home?lang=en&temp=default

No ApplicationController

```
public function initialize()
{
    parent::initialize();

    $this->loadComponent('RequestHandler');
    $this->loadComponent('Flash');

    $lang=$this->request->query('lang');
    $temp=$this->request->query('temp');

    $this->set('lang',$lang);
    $this->set('template',$temp);
}
```

Na View :

```
<div id="content">
  <div class="row">
    <div class="columns large-12 ctp-warning checks">
      <?php print "<h1>Idioma: ".$lang." Template: ".$template."</h1>"; ?>
```

Links:

<http://ribafs.org/cakephp/cake-control-demo/users/login?lang=en&temp=default>  
<http://ribafs.org/cakephp/cake-control-demo/users/login?lang=en&temp=bootstrap>  
<http://ribafs.org/cakephp/cake-control-demo/users/login?lang=pt&temp=default>  
<http://ribafs.org/cakephp/cake-control-demo/users/login?lang=en&temp=bootstrap>

## Controllers

Componentes de exemplo

```
<?php
/**
 * Component Calculo
 * For CakePHP 3.x
 * Autor: Ribamar FS
 *
 * This component allow access control to each action from application with web interface.
 *
 * Licenced with The MIT License
 * Redistributions require keep copyright below.
 *
 * @copyright Copyright (c) Ribamar FS (http://ribafs.org)
 * @link http://ribafs.org
 * @license http://www.opensource.org/licenses/mit-license.php MIT License
 */
```

```
namespace App\Controller\Component;
use Cake\Controller\Component;

class CalculoComponent extends Component{

    private $sum;
    private $dois;

    public function soma($sum,$dois){
        return ($sum + $dois);
    }

    public function subtracao($sum,$dois){
        return ($sum - $dois);
    }

    public function multiplicacao($sum,$dois){
        return ($sum * $dois);
    }

    public function divisao($sum,$dois){
        return ($sum / $dois);
    }

}
/*Usando
Salvar na pasta src/Controller/Component
```

Em um dos actions:

```
public $components = array('Calculo');
```

```
$soma = $this->Calculo->soma(2,3);
$multiplicacao = $this->Calculo->multiplicacao(2,3);
$this->set('soma',$soma);
$this->set('multiplicacao',$multiplicacao);
```

No Template correspondente:

```
<?php print 'A soma é '.$soma;?>
<?php print '<br>O produto é '.$multiplicacao;?>
*/
```

## Outro Component

```
<?php
/**
 * Component AccessControl
 * For CakePHP 3.x
 * Autor: Ribamar FS
 *
 * This component allow access control to each action from application with web interface.
 *
 * Licenced with The MIT License
 * Redistributions require keep copyright below.
 *
 * @copyright Copyright (c) Ribamar FS (http://ribafs.org)
 * @link http://ribafs.org
 * @license http://www.opensource.org/licenses/mit-license.php MIT License
 */
namespace App\Controller\Component;

use Cake\Controller\Component;

class AccessControlComponent extends Component{

    public $msg;
    public $components = array('Session','Auth');

    public function tables($controller){
        $this->Privilege = ClassRegistry::init('Privilege');

        // Database name
        App::uses('ConnectionManager', 'Model');
        $ds = ConnectionManager::getDataSource('default')->config;
        $db = $ds['database'];

        // All table names
        $tables = $this->Privilege->query("show tables");

        foreach($tables as $table){
            $table = $table['TABLE_NAMES']['Tables_in_'.$db];
            if($table != 'users' && $table != 'groups' && $table != 'privileges'){
```

```

        $action = $this->Privilege->query("select action from privileges
where group_name = 'managers' and controller = '$table'");
        if(!isset($action[0]['privileges']['action'])){
            $this->Privilege->query("insert into privileges
(group_name,controller,action) values ('managers', '$table', 'index')");
            $this->Privilege->query("insert into privileges
(group_name,controller,action) values ('managers', '$table', 'view')");
            $this->Privilege->query("insert into privileges
(group_name,controller,action) values ('managers', '$table', 'add')");
            $this->Privilege->query("insert into privileges
(group_name,controller,action) values ('managers', '$table', 'edit')");
            $this->Privilege->query("insert into privileges
(group_name,controller,action) values ('managers', '$table', 'delete')");
        }
    }
}

public function access($controller,$action){
    $this->Privilege = ClassRegistry::init('Privilege');// Allow model use in component

    $group = $this->Privilege->find('first', array(
'conditions'=>array('Privilege.controller'=>$controller,'Privilege.action'=>$action),
'fields'=>array('Privilege.group_name')
));

    if(isset($group['Privilege']['group_name'])){
        $group = $group['Privilege']['group_name'];
        return $group;
    }else{
        return false;
    }
}

public function go($controller,$action){
    $user = $this->Auth->user();
    $groupid=$user['group_id'];
    $access = $this->access($controller,$action);

    if($groupid == 2 && $access == 'admins'){// managers dont access admins

        return true;
    }elseif($access == false){// Privileges not registered
        $this->msg = 'priv';
        return false;
    }else{
        return false;
    }
}
}
}

```

## Criação de Component

Componentes são classes auxiliares dos controllers. Os Components podem ser considerados como formas de criar pedaços reutilizáveis de código relacionado a controllers com uma característica específica ou conceito. Os components também podem ligar-se ao evento do ciclo de vida do controller e interagir com a sua aplicação.

Criar o arquivo

CalcularComponent.php, contendo:

```
<?php
namespace App\Controller\Component;

use Cake\Controller\Component;

class CalcularComponent extends Component
{
    public function soma($valor1, $valor2)
    {
        $s = $valor1 + $valor2;
        return $s;
    }
}
```

E copiar para a pasta src/Controller/Component

Adicionar uma chamada no src/Controller/AppController

```
public function initialize()
{
    parent::initialize();
    $this->loadComponent('Auth');
    $this->loadComponent('RequestHandler');
    $this->loadComponent('Flash');
    $this->loadComponent('Calcular');
}
```

Executar num action, como o src/Controller/ClientesController.php

```
public function index()
{
    print 'Componente Soma: '.$this->Calcular->soma(2,3);
}
```

Estando no AppController ele será visto por todos os demais controllers e seus métodos estarão disponíveis logo que cada controller seja executado.

Caso queira que apenas um controller tenha acesso ao mesmo, insira uma chamada somente neste controller, assim:

```
$this->loadComponent('Calcular');
```

```
$soma = $this->Calcular->soma(2,4);
print 'Calcular: '.$soma;
```

## Models

### Behavior

#### Behaviors no CakePHP 3

`Cake\ORM\Table::addBehavior($name, array $options = [])`

Behaviors fornecem uma maneira fácil para criar peças de lógica reutilizáveis horizontal relacionados às classes Table. Você pode estar se perguntando por que behaviors/comportamentos são classes regulares e não traits/características. A principal razão para isso são os eventos de ouvintes/listeners. Enquanto traits permitiriam peças reutilizáveis de lógica, que iriam complicar eventos de ligação/bindings.

Para adicionar um comportamento/behavior para sua tabela você pode chamar o método `addBehavior()`. Geralmente o melhor lugar para fazer isso é no método `initialize()`:

```
namespace App\Model\Table;

use Cake\ORM\Table;

class ArticlesTable extends Table
{
    public function initialize(array $config)
    {
        $this->addBehavior('Timestamp');
    }
}
```

Assim como em associações, você pode usar a sintaxe de plugins e oferecer adicinas opções de configuração:

<http://book.cakephp.org/3.0/en/appendices/glossary.html#term-plugin-syntax>

```
namespace App\Model\Table;

use Cake\ORM\Table;

class ArticlesTable extends Table
{
    public function initialize(array $config)
    {
        $this->addBehavior('Timestamp', [
            'events' => [
                'Model.beforeSave' => [
                    'created_at' => 'new',
                    'modified_at' => 'always'
                ]
            ]
        ]);
    }
}
```

Você pode descobrir mais sobre os comportamentos/behaviors, incluindo os comportamentos/behaviors fornecidos pelo CakePHP no capítulo sobre Behaviors:

<http://book.cakephp.org/3.0/en/orm/behaviors.html>



Mais:

<http://book.cakephp.org/3.0/en/orm/table-objects.html>

## Entity

```
<?php
namespace App\Model\Entity;

use Cake\Auth\DefaultPasswordHasher;
use Cake\ORM\Entity;

class User extends Entity
{
    // Make all fields mass assignable except for primary key field "id".
    protected $_accessible = [
        '*' => true,
        'id' => false
    ];

    // ...

    protected function _setPassword($password)
    {
        return (new DefaultPasswordHasher)->hash($password);
    }

    // ...
}
?>
```

## Table

```
<?php
namespace App\Model\Table;

use Cake\ORM\Query;
use Cake\ORM\RulesChecker;
use Cake\ORM\Table;
use Cake\Validation\Validator;

/**
 * Users Model
 *
 * @property \Cake\ORM\Association\BelongsTo $Groups
 * @property \Cake\ORM\Association\HasMany $Clientes
 *
 * @method \App\Model\Entity\User get($primaryKey, $options = [])
 * @method \App\Model\Entity\User newEntity($data = null, array $options = [])
 * @method \App\Model\Entity\User[] newEntities(array $data, array $options = [])

```

```

* @method \App\Model\Entity\User|bool save(\Cake\Datasource\EntityInterface $entity, $options
= [])
* @method \App\Model\Entity\User patchEntity(\Cake\Datasource\EntityInterface $entity, array
$data, array $options = [])
* @method \App\Model\Entity\User[] patchEntities($entities, array $data, array $options = [])
* @method \App\Model\Entity\User findOrCreate($search, callable $callback = null, $options =
[])
*
* @mixin \Cake\ORM\Behavior\TimestampBehavior
*/
class UsersTable extends Table
{

/**
* Initialize method
*
* @param array $config The configuration for the Table.
* @return void
*/
public function initialize(array $config)
{
    parent::initialize($config);

    $this->setTable('users');
    $this->setDisplayField('id');
    $this->setPrimaryKey('id');

    $this->addBehavior('Timestamp');

    $this->belongsTo('Groups', [
        'foreignKey' => 'group_id',
        'joinType' => 'INNER'
    ]);
    $this->hasMany('Clientes', [
        'foreignKey' => 'user_id'
    ]);
}

/**
* Default validation rules.
*
* @param \Cake\Validation\Validator $validator Validator instance.
* @return \Cake\Validation\Validator
*/
public function validationDefault(Validator $validator)
{
    $validator
        ->integer('id')
        ->allowEmpty('id', 'create');

    $validator
        ->requirePresence('username', 'create')

```

```

->notEmpty('username')
->add('username', 'unique', ['rule' => 'validateUnique', 'provider' => 'table']);

$validator
->requirePresence('password', 'create')
->notEmpty('password');

return $validator;
}

/**
 * Returns a rules checker object that will be used for validating
 * application integrity.
 *
 * @param \Cake\ORM\RulesChecker $rules The rules object to be modified.
 * @return \Cake\ORM\RulesChecker
 */
public function buildRules(RulesChecker $rules)
{
    $rules->add($rules->isUnique(['username']));
    $rules->add($rules->existsIn(['group_id'], 'Groups'));

    return $rules;
}
}

```

### Implementação do displayField()

Tenho duas tabelas relacionadas: users e clientes.  
 Clientes tem um campo user\_id

Em  
 src/Model/Table/UsersTable.php

Mude o displayField para mostrar não o id mas o username

```
$this->displayField('username');
```

Aproveite e torne requerido o campo user\_id em clientes

src/Model/Table/ClientesTable.php

Adicione

```
$validator
->notBlank('user_id');
```

No Clientes/index.ctp no valor do user\_id:

```
<td><?= $cliente->has('user') ? $this->Html->link($cliente->user->id, ['controller' =>
'Users', 'action' => 'view', $cliente->user->id]) : " ?></td>
```

## Dicas de Validações

### Validações

#### E-mail requerido

```
$validator
->email('email')
->notBlank('email')
->add('email', 'unique', [
    'rule' => 'validateUnique',
    'provider' => 'table'
]);
```

ou

```
$validator
->requirePresence('email')
->add('email', 'validFormat', [
    'rule' => 'email',
    'message' => 'E-mail inválido'
]);
```

#### Nome requerido

```
$validator
->requirePresence('nome', 'create')
->notEmpty('nome');
```

#### No Model/Table/ClientesTable.php

#### Rule customizada para o campo nascimento

```
public function validationDefault(Validator $validator)
{
    $validator
        ->integer('id')
        ->allowEmpty('id', 'create');

    $validator
        ->requirePresence('nome', 'create')
        ->notEmpty('nome');

    $validator
        ->email('email')
        ->notBlank('email')
        ->add('email', 'unique', ['rule' => 'validateUnique', 'provider' => 'table']);

    $validator->add('nascimento',[
        'notEmptyCheck'=>[
            'rule'=>[$this,'notEmptyNascimento'],
            'provider'=>'table',
```

```

        'message'=>'Favor selecionar uma data de nascimento'
    ]
    });

    $validator
    ->allowEmpty('cpf');

    $validator
    ->allowEmpty('cnpj');

    $validator
    ->allowEmpty('fone');

    $validator
    ->allowEmpty('observacao');

    $validator
    ->notBlank('user_id');

    return $validator;
}

public function notEmptyNascimento($value,$context){
    if(empty($context['data']['nascimento'])) {
        return false;
    } else {
        return true;
    }
}
}

```

Dica: O campo group\_id não aparecia por padrão na lista de validações e mesmo que não fosse selecionado nenhum grupo o registro era armazenado. Então adicionei a validação como notBlank e na view add.ctp usei:

```

echo $this->Form->input('user_id', ['options' => $users,'empty'=>true]);

```

Assim aparece o asterisco vermelho indicando requerido e somente quando alguém escolhe um grupo na lista é permitido cadastrar.

Sugestão para lista

```

public function validationDefault(Validator $validator)
{
    return $validator
        ->notEmpty('first_name', 'A username is required')
        ->notEmpty('last_name', 'A username is required')
        ->notEmpty('email', 'A username is required')
        ->notEmpty('username', 'A username is required')
        ->notEmpty('password', 'A username is required')
        ->add('role', 'inList', [
            'rule' => ['inList', ['Employer', 'Job Seeker']],
            'message' => 'Please enter a valid role'
        ]);
}

```

```
    });  
}
```

Como o Cake exige que as datas sejam null por default, uma forma de contornar isso exigindo o preenchimento é criando uma validação customizada:

No Model/Table/ClientesTable.php

```
public function validationDefault(Validator $validator)  
{  
    ...  
    $validator->add('nascimento',[  
        'notEmptyCheck'=>[  
            'rule'=>[$this,'notEmptyNascimento'],  
            'provider'=>'table',  
            'message'=>'Favor selecionar uma data de nascimento'  
        ]  
    ]);  
    ...  
    return $validator;  
}  
  
public function notEmptyNascimento($value,$context){  
    if(empty($context['data']['nascimento'])) {  
        return false;  
    } else {  
        return true;  
    }  
}
```

Sugestão para lista

```
public function validationDefault(Validator $validator)  
{  
    return $validator  
        ->notEmpty('first_name', 'A username is required')  
        ->notEmpty('last_name', 'A username is required')  
        ->notEmpty('email', 'A username is required')  
        ->notEmpty('username', 'A username is required')  
        ->notEmpty('password', 'A username is required')  
        ->add('role', 'inList', [  
            'rule' => ['inList', ['Employer', 'Job Seeker']],  
            'message' => 'Please enter a valid role'  
        ]);  
}
```

## Dicas sobre Bancos de dados

Relacionamentos

O campo secundário do relacionamento precisa ser NOT NULL na tabela.

Exemplo  
users com clientes

Em clientes temos o campo user\_id que relaciona as tabelas.  
Na tabela clientes o campo user\_id precisa ser:  
user\_id int not null

Caso contrário teremos sérios problemas de cadastramento de todos os registros vazios e fura o relacionamento.

### **Perigo do Valor Default**

Na criação das tabelas não deixe valor default em campos.  
Caso deixe ele não será criticado, não será exigida uma entrada do usuário.  
Ao teclar enter sem ter digitado nada o valor default será armazenado.

Um bom exemplo é o campo user\_id no Clientes/add.ctp que por padrão é criado assim pelo Bake

```
echo $this->Form->input('user_id', ['options' => $users]);
```

Desta forma o primeiro valor é assumido sempre que o usuário não seleciona algo no select.

Idealmente mude para algo assim:

```
echo $this->Form->input('user_id', ['empty' => true,'options' => $users]);
```

Assim, ao teclar enter, sem ter selecionado nada será exigido que selecione algo.

## **15.10 - Form Helper**

Dicas sobre data e hora

Para o CakePHP os campos do tipo data, datetime ou timestamp obrigatoriamente devem usar DEFAULT NULL:

```
nascimento date DEFAULT NULL
```

E também não podemos alterar sua validação para exigir preenchimento com notEmpty ou notBlank. Caso contrário o Cake não reconhece e não adiciona o registro.

src/Template/Clientes/add.ctp ou edit.ctp

Ano mínimo sendo 13 anos antes do atual e máximo sendo 100 anos antes do atual, ou seja, como estou em 2016 de 1916 até 2003

```
echo $this->Form->input('nascimento',['label' => 'Nascimento',  
'dateFormat' => 'DMY',  
'minYear' => date('Y') - 100,
```

```

        'maxYear'    => date('Y') - 13,
        'empty'     => [
            'day'     => 'Dia',
            'month'   => 'Mês',
            'year'    => 'Ano'
        ]
    });

```

Select com valor default

```

$options = ['A'=>'Val1',
            'B'=>'Val2',
            'C'=>'Val3',
            'D'=>'Val4',
            'E'=>'Val5'];
print $this->Form->select('category', $options,['multiple'=>'checkbox', 'required'=>'false',
'label'=>'Category','class'=>'col-md-12','value'=>'E']);

```

Select múltiplo (permite selecionar várias opções)

```

print $this->Form->input('pilot_ratings',[
    'type' => 'select',
    'class' => 'listbox',
    'size' => 5,
    'id' => 'pilot_ratings',
    'multiple' => 'multiple',
    'options' => [
        ['name' => 'Habilitación de Vuelo Nocturno Local', 'value' => '1'],
        ['name' => 'Habilitación Cat. II / Cat. III', 'value' => '2'],
        ['name' => 'Habilitación de Remolque de Planeador', 'value' => '5']
    ]
]);

```

src/Template/Clientes/add.ctp ou edit.ctp

Ano mínimo sendo 13 anos antes do atual e máximo sendo 100 anos antes do atual, ou seja, como estou em 2016 de 1916 até 2003

```

        echo $this->Form->input('nascimento',['label' => 'Nascimento',
        'dateFormat' => 'DMY',
        'minYear' => date('Y') - 100,
        'maxYear' => date('Y') - 13,
        'empty' => [
            'day' => 'Dia',
            'month' => 'Mês',
            'year' => 'Ano'
        ]
    });

```



## HTML Helper

Para que Html->Link permita CSS:

```
<?= $this->Html->Link('<span class="glyphicon glyphicon-plus"></span> Novo',  
['controller'=>'Bookmarks','action'=>'add'],  
['class'=>'btn btn-primary pull-right']);
```

Assim ele mostrará o código <span...</span>

Para que permita o CSS, use:

```
<?= $this->Html->link('<span class="glyphicon glyphicon-plus"></span> Novo',  
['controller'=>'Bookmarks','action'=>'add'],  
['class'=>'btn btn-primary pull-right', 'escape'=>false]);
```

Observação:

classe Html, método link. Classe com inicial máiuscuça e método tudo em minúsculas.

## Form Helper

Mudar o tipo de um campo

O Cake gerou um campo com o tipo textarea em um form.

Para mudar podemos fazer isso:

```
print $this->Form->input('url',['label', 'URL']);
```

Mudar para tipo text (campo texto) assim:

```
print $this->Form->input('url',['type'=>'text','label', 'URL']);
```

## Helpers CakePHP 3

Os helpers contém código de apresentação que é compartilhado entre muitas views, elements ou layouts.

O Cake já traz alguns bem úteis Helpers:

- Flash
- Form
- Html
- Number
- Paginator
- Text
- Time
- Url

**Carregamos Helpers** no Cake declarando em classes views.

Class AppView extends View

```
{
    public function initialize()
    {
        parent::initialize();
        $this->loadHelper('Html');
        $this->loadHelper('Form');
        $this->loadHelper('Flash');
    }
}
```

### **Carregando Helpers em Plugins**

```
$this->loadHelper('ContactManager.Comentario');
```

Podemos carregar helper condicionalmente

class AppView extends View

```
{
    public function initialize()
    {
        parent::initialize();
        if ($this->request->action === 'index') {
            $this->loadHelper('Message');
        }
    }
}
```

### **Usando o Helper**

```
$this->Form->create($article);
```

Alguns helpers são carregados pelo Cake automaticamente, como é o caso do Form.

Também podemos usar o método `beforeRender()` dos controllers para carregar Helpers.

class ArticlesController extends ApplicationController

```
{
    public function beforeRender(Event $event)
    {
        parent::beforeRender($event);
        $this->viewBuilder()->helpers(['MyHelper']);
    }
}
```

### **Criando Helpers**

Podemos criar classes Helper para usar em aplicações ou em plugins.

Os Helpers têm algumas convenções que ajudam se seguirmos:

- Arquivos de classe Helper devem ficar em `src/View/Helper`. Exemplo:  
`src/View/Helper/LinkHelper.php`
- Classes helpers devem ser sufixadas com Helper. Exemplo:  
`LinkHelper`
- Quando referenciar classes helper deve omitir o sufixo Helper:

```
$this->loadHelper('Link');
```

Criando o esqueleto de um Helper chamado Message com o bake:

```
cd c:\xampp\htdocs\aplicativo
bin\cake bake helper Message
```

Editar o src\View\Helper\MessageHelper.php e adicionar as funções, assim:

```
<?php
namespace App\View\Helper;

use Cake\View\Helper;
use Cake\View\View;

/**
 * Table helper
 */
class TableHelper extends Helper
{

    /**
     * Default configuration.
     *
     * @var array
     */
    protected $_defaultConfig = [];

    public function msg($msg)
    {
        return '<h2>'.$msg.'</h1>';
    }
}
```

### Carregar na classe AppView

Editar src/View/AppView.php e adicionar o loadHelper()

```
class AppView extends View
{
    public function initialize()
    {
        parent::initialize();
        if ($this->request->action === 'index') {
            $this->loadHelper('Message');
        }
    }
}
```

Atente para o fato de que na classe acima ele restringe para que somente no action index.ctp o helper apareça. Para mostrar em todas as actions use:

**Observação** – pelos meus testes nem precisa carregar na AppView e o helper ficará disponível automaticamente.

```

public function initialize()
{
    $this->loadHelper( 'Message' );
}

```

## Usando o Helper

Editar a view index.ctp e adicionar:

```
<?=$this->Message->msg('Minha Mensagem Helper') ?>
```

Ao chamar no navegador verá a mensagem com título h2.  
Um helper bem simples para mostrar o caminho das pedras.

Documentação oficial:

<http://book.cakephp.org/3.0/en/views/helpers.html>

## Templates

### Criação de Menu com Element

Este element deve funcionar apenas para aplicativos que aplicaram o plugin Acl e seus usuários chamam-se:

admin, manager e user. Caso sejam diferentes faça as devidas alterações.

Abrir o AppController.php

Adicionar as linhas seguintes dentro do método initialize()

```

$loguser = $this->request->session()->read('Auth.User');
$loguser = $loguser['username'];
$this->set('loguser',$loguser);

```

Criar o arquivo topmenu.ctp contendo:

```

<?php
$loggedno = 'Não Logado';
if(!$loguser) $loguser=$loggedno;

if($loguser == 'admin'){
    echo $this->Html->link(__('Clientes'),
array('plugin'=>null,'controller'=>'Clientes','action'=>'index'));
    echo '&nbsp;'.$this->Html->link(__('Funcionários'),
array('plugin'=>null,'controller'=>'Funcionarios','action'=>'index'));
    echo '&nbsp;'.$this->Html->link(__('Grupos'),
array('plugin'=>null,'controller'=>'Groups','action'=>'index'));
    echo '&nbsp;'.$this->Html->link(__('Usuários'),
array('plugin'=>null,'controller'=>'Users','action'=>'index'));
    echo '&nbsp;'.$this->Html->link(__('Sair'),
array('plugin'=>null,'controller'=>'Users','action'=>'logout'));
}elseif($loguser == 'manager'){

```



```

?>
</head>
<body>

<!-- If you'd like some sort of menu to
show up on all of your views, include it here -->
<div id="header">
  <div id="menu">...</div>
</div>

<!-- Here's where I want my views to be displayed -->
<?= $this->fetch('content') ?>

<!-- Add a footer to each displayed page -->
<div id="footer">...</div>

</body>
</html>

```

Podemos definir vários layouts para nosso aplicativo.

Os layouts devem ficar no diretório  
src/Template/Layout

O CakePHP já vem com um layout default.ctp.

Atribuindo título para o aplicativo e definindo um layout no ApplicationController:

```

class UsersController extends ApplicationController
{
  public function view_active()
  {
    $this->set('title', 'View Active Users');
    $this->viewBuilder()->layout('default_small_ad');
  }

  public function view_image()
  {
    $this->viewBuilder()->layout('image');
  }
}

```

View add.ctp

```

<div class="row">
  <div class="col-xs-12">
    <div class="actions">
      <ul class="side-nav btn-group">
        <li class="btn btn-info btn-sm"><?= $this->Html->link(__('List Users'), ['action' =>
'index']) ?></li>
      </ul>
    </div>
  </div>
</div>

```

```

    </div>
  </div>
</div>
<div class="row">
  <div class="col-xs-12">
    <div class="users form">
      <?= $this->Form->create($user, ['role' => 'form']) ?>
      <div class="box box-primary">
        <div class="box-header with-border">
          <h3 class="box-title"><?= __('Add User') ?></h3>
        </div>
        <div class="box-body">
          <div class="form-group">
            <?php
              echo $this->Form->input('username', ['class' => 'form-control', 'placeholder' =>
                __('Enter ...')]);
              echo $this->Form->input('password', ['class' => 'form-control', 'placeholder' =>
                __('Enter ...')]);
              echo $this->Form->input('email', ['class' => 'form-control', 'placeholder' => __('Enter
                ...')]);
              echo $this->Form->input('name', ['class' => 'form-control', 'placeholder' => __('Enter
                ...')]);
              echo $this->Form->input('surname', ['class' => 'form-control', 'placeholder' =>
                __('Enter ...')]);
            ?>
          </div>
        </div>
        <div class="box-footer">
          <?= $this->Form->button(__('Submit'), ['class'=>'btn btn-primary']) ?>
          <?= $this->Html->link(__('Cancel'), ['controller' => 'users'], ['class' => 'btn btn-
            warning']) ?>
        </div>
      </div>
      <?= $this->Form->end() ?>
    </div>
  </div>
</div>

```

## 15.11 - Um Resumo da codificação do CakePHP

(camadas Model, Controller e View/Template)

```
class ArticlesController extends AppController
{
    public function index()
    {
        $this->set('articles', $this->Articles->find('all'));
    }

    public function view($id = null)
    {
        $article = $this->Articles->get($id);
        $this->set(compact('article'));
    }
}
```

Repare que você está usando get() ao invés de find('all') porquê nós queremos a informação de apenas um artigo.

Repare que nossa action recebe um parâmetro: o ID do artigo que gostaríamos de visualizar.

Se o usuário requisitar /articles/view/3, então o valor '3' é passado como \$id para a action.

**Para receber segurança com strings nos forms, usar a função h():**

```
<!-- File: src/Template/Articles/view.ctp -->
```

```
<h1><?= h($article->title) ?></h1>
<p><?= h($article->body) ?></p>
```

**Testar se uma requisição é realmente post:**

```
if ($this->request->is('post')) {
```

**Debugar**

```
debug($variavel);exit;
```

**Métodos success e error do FlashComponent para definir mensagens**

```
if ($this->Articles->save($article)) {
    $this->Flash->success(__('Seu artigo foi salvo.));
    return $this->redirect(['action' => 'index']);
}
$this->Flash->error(__('Não é possível adicionar o seu artigo.));
```



## Validação nos Forms

Para tirar proveito dos recursos de validação do Cake, você vai precisar usar o Form helper em suas views.

Basta usar `$this->Form->`.

Exemplo de view add:

```
<!-- File: src/Template/Articles/add.ctp -->
<h1>Add Article</h1>
<?php
    echo $this->Form->create($article);
    echo $this->Form->input('title');
    echo $this->Form->input('body', ['rows' => '3']);
    echo $this->Form->button(__('Salvar artigo'));
    echo $this->Form->end();
?>
```

## Regras de validação são definidas no Model, exemplo

```
public function validationDefault(Validator $validator)
{
    $validator
        ->notEmpty('title')
        ->notEmpty('body');

    return $validator;
}

if ($this->request->is(['post', 'put'])) {
    $this->Articles->patchEntity($article, $this->request->getData());
}
```

Em seguida, a action verifica se a requisição é POST ou PUT e caso seja, os dados são usados para atualizar a entidade de artigo em questão ao usar o método `patchEntity()`.

## Recebendo o nome de um action

```
$this->request->getParam('action')
```

## Recebendo nome de Controller

```
$controllerName = $this->request->getParam('controller');
```

## Camada Model

```
use Cake\ORM\TableRegistry;

$users = TableRegistry::get('Users');
$query = $users->find();
foreach ($query as $row) {
    echo $row->username;
}
```

**Se nós quiséssemos criar um usuário e salvá-lo (com validação) faríamos algo assim:**

```
use Cake\ORM\TableRegistry;

$users = TableRegistry::get('Users');
$user = $users->newEntity(['email' => 'mark@example.com']);
$users->save($user);
```

### **A camada view**

// No arquivo view, nós renderizaremos um 'elemento' para cada usuário.

```
<?php foreach ($users as $user): ?>
    <div class="user">
        <?= $this->element('user', ['user' => $user]) ?>
    </div>
<?php endforeach; ?>
```

### **Controller**

Um controller pode ser visto como um gerente que certifica-se que todos os recursos necessários para completar uma tarefa sejam delegados aos trabalhadores corretos.

Ele aguarda por petições dos clientes, checa suas validades de acordo com autenticação ou regras de autorização, delega requisições ou processamento de dados da camada Model, selecciona o tipo de dados de apresentação que os clientes estão aceitando, e finalmente delega o processo de renderização para a camada View.

Enviar informações do controller para uma view:

```
$this->set('valores', $valores);
```

### **Somente acaitar DELETE ou POST**

```
$this->request->allowMethod(['post', 'delete']);
```

### **Enviar informações da view para o controller**

in your view you can use get method to send the form data. and in your controller's action you can access the passed arguments with `$this->params['url'];`

in your view

```
$this->Form->create('Model', array('type' => 'get', 'action' => 'search'));
$this->Form->input('select_tfield_id', array('type' => 'select'));
$this->Form->input('value');
$this->Form->end('submit');
```

In your controller

```
function search() {
```

```
$url = $this->params['url'];
$id = $url['select_tfield_id'];
$value = $url['value'];
}
```

Pelo form

```
<?= $this->Form->hidden('name', ['value' => $this->Form->select('name')]) ?>
```

Não passa nada, apenas null

### Carregar componentes

Usar no método initialize()

```
$this->loadComponent('Flash');
$this->loadComponent('Auth');
```

### Alguns actions típicos:

```
public function login()
{
    if ($this->request->is('post')) {
        $user = $this->Auth->identify();
        if ($user) {
            $this->Auth->setUser($user);
            return $this->redirect($this->Auth->redirectUrl());
        }
        $this->Flash->error('Your username or password is incorrect.');
```

```
    }
}

public function logout()
{
    $this->Flash->success('You are now logged out.');
```

```
    return $this->redirect($this->Auth->logout());
}

public function index()
{
    $this->set('articles', $this->Articles->find('all'));
}

public function view($id)
{
    $article = $this->Articles->get($id);
    $this->set(compact('article'));
}

public function add()
{
    $article = $this->Articles->newEntity();
    if ($this->request->is('post')) {
```

```

        $article = $this->Articles->patchEntity($article, $this->request->getData());
        if ($this->Articles->save($article)) {
            $this->Flash->success(__('Your article has been saved.));
            return $this->redirect(['action' => 'index']);
        }
        $this->Flash->error(__('Unable to add your article.));
    }
    $this->set('article', $article);
}

```

// src/Controller/ArticlesController.php

```

public function edit($id = null)
{
    $article = $this->Articles->get($id);
    if ($this->request->is(['post', 'put'])) {
        $this->Articles->patchEntity($article, $this->request->getData());
        if ($this->Articles->save($article)) {
            $this->Flash->success(__('Your article has been updated.));
            return $this->redirect(['action' => 'index']);
        }
        $this->Flash->error(__('Unable to update your article.));
    }

    $this->set('article', $article);
}

```

// src/Controller/ArticlesController.php

```

public function delete($id)
{
    $this->request->allowMethod(['post', 'delete']);

    $article = $this->Articles->get($id);
    if ($this->Articles->delete($article)) {
        $this->Flash->success(__('The article with id: {0} has been deleted.',
h($id)));
        return $this->redirect(['action' => 'index']);
    }
}

```

## Algumas views típicas:

```
<!-- File: src/Template/Articles/add.ctp -->
```

```
<h1>Add Article</h1>
<?php
    echo $this->Form->create($article);
    echo $this->Form->control('title');
    echo $this->Form->control('body', ['rows' => '3']);
    echo $this->Form->button(__('Save Article'));
    echo $this->Form->end();
?>
```

```
<!-- File: src/Template/Articles/edit.ctp -->
```

```
<h1>Edit Article</h1>
<?php
    echo $this->Form->create($article);
    echo $this->Form->control('title');
    echo $this->Form->control('body', ['rows' => '3']);
    echo $this->Form->button(__('Save Article'));
    echo $this->Form->end();
?>
```

```
<!-- File: src/Template/Articles/view.ctp -->
```

```
<h1><?= h($article->title) ?></h1>
<p><?= h($article->body) ?></p>
<p><small>Created: <?= $article->created->format(DATE_RFC850) ?></small></p>
```

```
<!-- File: src/Template/Articles/index.ctp (edit links added) -->
```

```
<!-- File: src/Template/Articles/index.ctp (delete links added) -->
```

```
<h1>Blog articles</h1>
<p><?= $this->Html->link('Add Article', ['action' => 'add']) ?></p>
<table>
    <tr>
        <th>Id</th>
        <th>Title</th>
        <th>Created</th>
        <th>Actions</th>
    </tr>
```

```
<!-- Here's where we loop through our $articles query object, printing out article info -->
```

```
<?php foreach ($articles as $article): ?>
<tr>
    <td><?= $article->id ?></td>
    <td>
        <?= $this->Html->link($article->title, ['action' => 'view', $article->id]) ?>
```

```

</td>
<td>
    <?= $article->created->format(DATE_RFC850) ?>
</td>
<td>
    <?= $this->Form->postLink(
        'Delete',
        ['action' => 'delete', $article->id],
        ['confirm' => 'Are you sure?'])
    ?>
    <?= $this->Html->link('Edit', ['action' => 'edit', $article->id]) ?>
</td>
</tr>
<?php endforeach; ?>

</table>

<!-- File: src/Template/Users/login.ctp -->

<div class="users form">
<?= $this->Flash->render() ?>
<?= $this->Form->create() ?>
    <fieldset>
        <legend><?= __('Please enter your username and password') ?></legend>
        <?= $this->Form->control('username') ?>
        <?= $this->Form->control('password') ?>
    </fieldset>
    <?= $this->Form->button(__('Login')); ?>
    <?= $this->Form->end() ?>
</div>

```

### Configurando a route:

```
$routes->connect('/', ['controller' => 'Pages', 'action' => 'display', 'home']);
```

para

```
$routes->connect('/', ['controller' => 'Articles', 'action' => 'index']);
```

### Select num Controller

```
use Cake\ORM\TableRegistry;
```

```

$query = TableRegistry::get('Articles')->find();

foreach ($query as $article) {
    debug($article->title);
}

```

## Por default uma consulta traz todos os campos de uma tabela

```
$query = $articles->find();  
$articles->find()->select();
```

Para trazer apenas id e title:

```
$articles->find()->select(['id', 'title']);
```

## Retornar todos os title

```
use Cake\ORM\TableRegistry;  
$articles = TableRegistry::get('Articles');  
$allTitles = $articles->find()->extract('title');  
foreach ($allTitles as $title) {  
    echo $title.'  
';  
}
```

Par chave-valor:

```
$list = $articles->find('list');
```

```
foreach ($list as $id => $title) {  
    echo "$id : $title"  
}
```

## Retornando os title

```
$query = TableRegistry::get('Articles')->find();
```

```
foreach ($query as $article) {  
    debug($article->title);  
}
```

## Retornando um único registro de uma tabela

```
$articles = TableRegistry::get('Articles');  
$article = $articles  
    ->find()  
    ->where(['id' => 1])  
    ->first();
```

```
debug($article->title);
```

## Mostrar apenas um campo (title)

```
$articles = TableRegistry::get('Articles');  
$query = $articles->find();  
$query->select(['id', 'title', 'body']); // ou apenas $query->select();  
foreach ($query as $row) {  
    debug($row->title);  
}
```

### Limitar registros e páginas:

```
// Fetch rows 50 to 100
$query = $articles->find()
    ->limit(50)
    ->page(2);
```

### Retornar todo o primeiro registro:

```
$articles = TableRegistry::get('Articles');
$query = $articles->find('all', [
    'order' => ['Articles.created' => 'DESC']
]);
$row = $query->first();
print $row;
```

### Métodos Dinâmicos encontrar

```
$query = $this->Users->findByUsername('usuario');
$query = $this->Users->findAllByUsername('usuario');
```

### Encontrar relacionados

```
$query = $articles->find()->contain([
    'Authors' => ['Addresses'], 'Comments' => ['Authors']
]);

$query = $products->find()->contain([
    'Shops.Cities.Countries',
    'Shops.Managers'
]);
```

### Inserir um novo registro

```
$articlesTable = TableRegistry::get('Articles');
$article = $articlesTable->newEntity();

$article->title = 'A New Article';
$article->body = 'This is the body of the article';

Atualizar
$articlesTable = TableRegistry::get('Articles');
$article = $articlesTable->get(12); // Return article with id 12

$article->title = 'CakePHP is THE best PHP framework!';
$articlesTable->save($article);
```



## 15.12 - Comunicação entre Model, Controller e View

### Criar function no model:

```
public function despesasTotal(){
    $query = $this->find('all'); //fetch the record
    $res = $query->select(['total_val' =>$query->func()->sum('valor_unit')]->first();
    $total = $res->total_val;
    return $total;
}
```

Alternativa:

```
use Cake\Datasource\ConnectionManager;
```

```
public function despesasTotal(){
    $connection = ConnectionManager::get('default');
    $results = $connection->execute('SELECT sum(valor_unit) FROM despesas')-
>fetchAll();
    return $results;
}
```

### No controller/action:

```
use Cake\ORM\TableRegistry;
```

```
public function total()
{
    $despesas = TableRegistry::get('Despesas');
    $totais = $despesas->despesasTotal();
    $this->set('totais',$totais);
}
```

### Na view/template

```
foreach($totais as $total){
    foreach($total as $tot){
        print $tot;
    }
}
```

### Resumindo:

- o model traz do banco de dados
- o controller traz do model
- o template/view traz do controller

## Contagem e outros

```
$query = $this->Model->find()
->where([
    'category_id'=>1,
    'ativo'=>1
])->order('created DESC');
```

```
debug($query->all());
debug($query->first());
debug($query->count());
```

## Query Builder

```
$query = $this->Model->find()
->where([
    'category_id'=>1,
    'ativo'=>1
])
->order('created DESC')
->contain([
    'Category'
]);
```

//ou com get()

```
$data = $this->Model->get(1, [
    'contain'=>['Category']
]);
```

## 15.13 - Customizar paginação

```
class UsersController extends ApplicationController
{
    private $roles = ['User' => 'User', 'Admin' => 'Admin', 'Disabled' => 'Disabled'];
    public $paginate = [
        'limit' => 30,
        'order' => ['first_name' => 'ASC', 'last_name' => 'ASC']
    ];
    public function index() {
        $users = $this->paginate($this->Users);
        $this->set(compact('users'));
        $this->set('_serialize', ['users']);
    }
}
```

## Usando Elements

= Criar um ou mais elements. No exemplo tenho 3: head, header e footer

Arquivos:

src/Template/Element/nome.ctp

= Criar um novo Layout e encaixar os elements com `<?= $this->element('nome') ?>`:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <?= $this->element('head') ?>
</head>
<body>
  <?= $this->element('header') ?>
  <!-- Page Content -->
  <div id="content" class="container">
    <?= $this->Flash->render() ?>
    <div class="row">
      <?= $this->fetch('content') ?>
    </div>
  </div>
  <?= $this->element('footer') ?>
</body>
</html>
```

= Adicionar à função initialize() do Controller:

```
$this->viewBuilder()->layout('nome_layout');
```

## Carregar model que não é o default

```
$this->loadModel('Clientes');
$receitas = $this->Clientes->metodo_cliente();
```

## 15.14 - Acesso a Banco de Dados

O trabalho com bancos de dados no Cake é feito com dois objetos Tables (lida com coleções da dados, tabela, por exemplo) e Entities (lida com apenas um registro).

Para trabalhar com Tabelas num controller

Carregar o objeto Table

```
use Cake\ORM\TableRegistry;
```

Carregar o respectivo objeto

```
$clientes = TableRegistry::get('Clientes');
```

Agora pode trabalhar com seu conteúdo.

## Acesso a Bancos de Dados

```
use Cake\Datasource\ConnectionManager;
```

```
$dsn = 'mysql://root:password@localhost/my_database';  
ConnectionManager::config('default', ['url' => $dsn]);  
$conn = ConnectionManager::get('default');
```

```
$results = $conn->execute('SELECT * FROM articles')->fetchall('assoc');
```

Também podemos usar query builder.

Datales - <https://book.cakephp.org/3.0/en/orm/database-basics.html>

## Usando transações

```
$conn->begin();  
$conn->execute('UPDATE articles SET published = ? WHERE id = ?', [true, 2]);  
$conn->execute('UPDATE articles SET published = ? WHERE id = ?', [false, 2]);  
$conn->commit();
```

## 15.15 - Usando displayField()

Neste caso tenho duas tabelas relacionadas, clientes e users. A view index de clientes mostra o campo de users, mas somente o id. Quero mudar para que mostra o username.

Código criado pelo bake deixa campos relacionados assim:

### Clientes

Id	Name	Birthday	Phone	User
10	Rudyard Web er	10/26/15	(445) 457-45 52	1
13	Dustin M. On eill	4/24/16	(276) 722-09 76	1
19	Wesley Garne r	6/11/16	(578) 231-23 89	1
20	Irene P. Arnol d	2/12/16	(253) 631-98 30	1
22	Roanna K. Or tiz	8/28/16	(336) 788-21 15	1

Onde o User aparece como código.

Para que mostre o username, apenas altere o src/Template/Cientes/index.ctp na linha:

```
<td><?=$cliente->has('user') ? $this->Html->link($cliente->user->id, ['controller' => 'Users', 'action' => 'view', $cliente->user->id]) : " ?></td>
```

Mude id para username assim:

```
<td><?=$cliente->has('user') ? $this->Html->link($cliente->user->username, ['controller' => 'Users', 'action' => 'view', $cliente->user->id]) : " ?></td>
```

Então a captura mostra

## Cientes

Id	Name	Birthday	Phone	User
10	Rudyard Weber	10/26/15	(445) 457-4552	super
13	Dustin M. O'Neill	4/24/16	(276) 722-0976	super
19	Wesley Garner	6/11/16	(578) 231-2389	super
20	Irene P. Arnold	2/12/16	(253) 631-9830	super
22	Roanna K. Ortiz	8/28/16	(336) 788-2145	super